공학박사학위논문

DNN-based Acoustic Modeling for Robust Automatic Speech Recognition

강인한 음성인식을 위한 DNN 기반 음향 모델링

2019년 2월

서울대학교 대학원

전기 · 컴퓨터공학부

이 강 현

Abstract

In this thesis, we propose three acoustic modeling techniques for robust automatic speech recognition (ASR). Firstly, we propose a DNN-based acoustic modeling technique which makes the best use of the inherent noise-robustness of DNN is proposed. By applying this technique, the DNN can automatically learn the complicated relationship among the noisy, clean speech and noise estimate to phonetic target smoothly. The proposed method outperformed noise-aware training (NAT), i.e., the conventional auxiliary-feature-based model adaptation technique in Aurora-5 DB.

The second method is multi-channel feature enhancement technique. In the general multi-channel speech recognition scenario, the enhanced single speech signal source is extracted from the multiple inputs using beamforming, i.e., the conventional signal-processing-based technique and the speech recognition process is performed by feeding that source into the acoustic model. We propose the multi-channel feature enhancement DNN algorithm by properly combining the delay-and-sum (DS) beamformer, which is one of the conventional beamforming techniques and DNN. Through the experiments using multichannel wall street journal audio visual (MC-WSJ-AV) corpus, it has been shown that the proposed method outperformed the conventional multi-channel feature enhancement techniques.

Finally, uncertainty-aware training (UAT) technique is proposed. The most of

the existing DNN-based techniques including the techniques introduced above, aim to optimize the point estimates of the targets (e.g., clean features, and acoustic model parameters). This tampers with the reliability of the estimates. In order to overcome this issue, UAT employs a modified structure of variational autoencoder (VAE), a neural network model which learns and performs stochastic variational inference (VIF). UAT models the robust latent variables which intervene the mapping between the noisy observed features and the phonetic target using the distributive information of the clean feature estimates. The proposed technique outperforms the conventional DNN-based techniques on Aurora-4 and CHiME-4 databases.

Keywords: Robust speech recognition, feature enhancement, feature compensation, acoustic modeling, deep neural network (DNN), variational autoencoder (VAE), variational inference (VIF), uncertainty decoding (UD)

Student number: 2012-20822

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, deep learning techniques have grown prevalent in the field of signal processing research, which continuously provided venues for drastic improvements in solving automatic speech recognition (ASR) tasks. In acoustic modeling, in particular, the introduction of the deep neural network (DNN)-hidden Markov model (HMM) framework, which exploits DNN instead of the conventional Gaussian mixture model (GMM) in order to compute the likelihood of the HMM states, has proven to be a breakthrough [1], [2]. Its capability to automatically learn the complicated non-linear relation between the input and the target vector has placed DNN as one of the most dominant approaches in robust ASR.

DNN-based approaches to robust ASR can generally be categorized into two types: feature-based and model-based techniques. The feature-end techniques [3]–[6] train a DNN by directly mapping the corrupted speech features to their clean counterparts, whereas other conventional techniques require the signal corruption process to be formulated into a specific model. The featureont-end techniques using DNN has shown outstanding performance in reconstructing clean features from the

noisy ones. The joint training strategy, in which acoustic and the feature processing DNNs are jointly optimized via concatenation, further improved performance.

The model-based techniques [7]–[12], on the other hand, rely on DNN parameters for automatically learning the mapping from the observed noisy speech to the phonetic targets, while the actual observations remain unaltered. These techniques, then, call for a carefully designed strategy to incorporate the environmental characteristics as the DNN-based acoustic model learns relevant parameters. Among various approaches, adaptation techniques employing auxiliary features with acoustic context information have shown impressive performance in robust ASR. These techniques enhance the performance of the acoustic model by augmenting additional information (e.g., background noise estimate and speaker information) to the input or target vector in order to improve the modeling power of the DNN. As an example, the technique referred to as noise-aware training (NAT) attained the notable results on Aurora-4 task [10]. NAT enables the DNN to learn the relationship among noisy input, noise features and target vectors corresponding to the phonetic identity by augmenting an estimate of the noise present in the input signal. As a result, although these two approaches are different in the detailed method they are same in perspective of aiming to mitigate the input data and trained acoustic model. Especially, when DNNs are introduced in both feature- and model-based techniques, the two DNNs can be seen as single larger network which performs the acoustic modeling.

In this thesis, DNN-based acoustic modeling techniques for robust ASR are proposed. In Chapter 3, we propose a technique which helps the DNN to address the complicated connection between the input and target vectors of NAT smoothly. The main idea of the proposed approach is to let the DNN clarify the relationship among noisy features, noise estimates and phonetic targets only after reconstruct-

ing the clean features. In order to accomplish this, the proposed technique cascades two individually fine-tuned DNNs into a single DNN and training the unified DNN jointly. The first DNN performs reconstruction of the clean features from noisy features when noise estimates are augmented. Then the next DNN attempts to learn the mapping between the reconstructed features and the phonetic targets. It has been shown that the proposed technique outperforms the conventional DNN-based techniques on Aurora5-task [13] and mismatched noise conditions.

While the above DNN-based techniques targets the close-talking scenario where the distance between the speaker and microphone is close, a multi-channel-based feature mapping technique is proposed in Chapter 4. In the general multi-channel speech recognition scenario, the enhanced single speech signal source is extracted from the multiple inputs using beamforming, i.e., the conventional signal-processing-based technique and the speech recognition process is performed by feeding that source into the acoustic model. The proposed multi-channel feature enhancement DNN algorithm combines the delay-and-sum (DS) beamformer, which is one of the conventional beamforming techniques and DNN. By this way, the proposed technique models the complicated relationship between the array inputs and clean speech features effectively by employing intermediate target. Through the experiments using multichannel wall street journal audio visual (MC-WSJ-AV) corpus [14], it has been shown that the proposed method outperformed the conventional multi-channel feature enhancement techniques.

Although these conventional DNN-based techniques have shown better performances, there still exists the limitation of them. The conventional DNN-based techniques aim to obtain the optimal point estimates of the target such as clean features and model parameters. So the estimated clean features or the phonetic targets may

still be unreliable due to various sources of uncertainty. Yet, these sources of uncertainty are mostly overlooked when applying DNN-based techniques, which eventually tampers with model performance. When the test data contains unseen environmental effects (e.g., noise, reverberation, and speaker and channel mismatch) which are seldom observed in the training data, the accuracy of the estimator decreases and this degrades the overall performance of the ASR system.

In Chapter 5, we propose a deep learning-based acoustic modeling technique which systematically measures and takes account of the uncertainty inherent in the input features using a single deep network. Our proposed technique, the uncertainty-aware training (UAT), namely, employs variational autoencoder (VAE), one of the widely used variational inference (VIF) techniques, which allows the extraction of robust features along with the associated uncertainties. VAE performs efficient inference under the assumption that the observed data is generated from a random variable. UAT modifies both the input and output structures of VAE so as to take the full advantage of DNN-based approach with auxiliary features, a structure similar to those introduced. UAT provides robust latent variables which intervene the mapping between the noisy observed features and the phonetic target by using the distributive information of the clean feature estimates. The proposed technique, along with the conventional DNN-based techniques, is evaluated on Aurora-4 and CHiME-4 databases [15]. Experimental results show that the proposed technique outperforms the conventional DNN-based techniques. Moreover, we confirm that the latent variables obtained from the proposed technique can be utilized as an effective measure of uncertainty.

The rest of the thesis is organized as follows: The next chapter introduces the basic structure of the DNN and the experimental database used in this thesis. In Chap-

ter 3, a DNN-based acoustic modeling technique for noise-robust ASR is proposed. In Chapter 4, DNN-based feature enhancement for robust multichannel speech recognition is introduced. Finally, a uncertainty-aware training for DNN-HMM system using variational inference is proposed in Chapter 5. The conclusions are drawn in Chapter 6.

Chapter 2

Background

This chapter presents some background for the research presented in this thesis. Firstly, we introduces DNN, which is the key algorithm of DNN-HMM system and the thesis. Also, various databases (DBs) used for evaluating the proposed techniques are described.

2.1 Deep Neural Networks

DNN is a multi-layer perceptron network with many hidden layers. A DNN consists of input, hidden and output layers as shown in Fig. 2.1. For simplicity, we denote the input layer as layer 0 and the output layer as layer L for an $(L+1)$-layer DNN.

The hidden representation of the DNN at the l-th layer can be written by

$$\mathbf{v}^l = \sigma(\mathbf{z}^l) = \sigma(\mathbf{W}^l \mathbf{v}^{l-1} + \mathbf{b}^l), \text{ for } 0 < l < L \tag{2.1}$$

where $\mathbf{v}^l = [v_1^l \ v_2^l \ \cdots \ v_{N_l}^l]'$, $\mathbf{z}^l = \mathbf{W}^l \mathbf{v}^{l-1} + \mathbf{b}^l = [z_1^l \ z_2^l \ \cdots \ z_{N_l}^l]'$, $\mathbf{W}^l$, $\mathbf{b}^l = [b_1^l \ b_2^l \ \cdots \ b_{N_l}^l]'$ and N_l denote the activation vector, excitation vector, weight ma-

trix with size $N_l \times N_{l-1}$, bias vector and the number of neurons at the l-th layer, respectively. Here, the prime denotes the transpose of a vector or a matrix. In (2.1), $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function which is usually employed as an activation function in many applications. The function $\sigma(\cdot)$ is applied to the excitation vector element-wisely. At the 0-th layer, $\mathbf{v}^0 = [v_1^0 \; v_2^0 \; \cdots \; v_{N_0}^0]'$ is the input vector and N_0 is the input feature dimension.

The data type at the output layer is decided based on the target task. For a multi-class classification task, each output neuron represents a class membership for which the softmax function is applied to $\mathbf{z}^L$ as follows:

$$v_i^L = \text{softmax}_i(\mathbf{z}^L) = \frac{e^{z_i^L}}{\sum_{j=1}^{N_L} e^{z_j^L}} \tag{2.2}$$

$$\sum_{i=1}^{N_L} v_i^L = 1 \tag{2.3}$$

where v_i^L, z_i^L and N_L indicate the i-th component of the output activation, the i-th component of the excitation vector and the number of classes at the output layer, respectively.

For supervised fine-tuning, a labeled training set $(\mathbf{o}, \mathbf{d}) = \{(o_t, d_t)|1 \le t \le T\}$ is needed where o_t represents the t-th observation vector, $d_t = [d_{t,1} \; d_{t,2} \; \cdots \; d_{t,N_L}]'$ is the corresponding target vector with size N_L and T denotes the number of training samples. The DNN input $\mathbf{v}_t^0 = [v_{t,1}^0 \; v_{t,2}^0 \; \cdots \; v_{t,N_0}^0]'$ at time t usually consists of a number of concatenated observation vectors. During fine-tuning, the DNN parameters are updated by using the back-propagation procedure according to a proper objective function. For multi-class classification, the cross-entropy (CE) is usually adopted as an objective function as given by

$$J_{CE} = \frac{1}{T} \sum_{t=1}^{T} \left[-\sum_{i=1}^{N_L} d_{t,i} \log(v_{t,i}^L) \right] \tag{2.4}$$

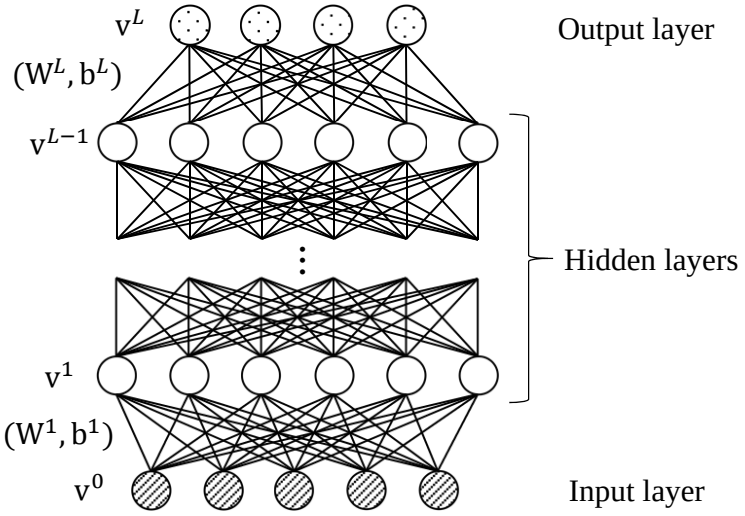


Figure 2.1: The structure of DNN.

where $d_{t,i}$ and $v_{t,i}^L$ indicate the i-th component of the desired target value and the i-th component of the generated DNN output value given the t-th observation. Basically, $d_{t,i}$ can be regarded as the posterior probability of the i-th output class.

2.2 Experimental Database

In this thesis, the four different DBs are used: Aurora-4 DB [16], Aurora-5 DB [13], MC-WSJ-AV DB [14] and CHiME-4 DB [15].

For that, we choose two kinds of DBs widely used in robust speech recognition area: Aurora-4 and Aurora-5 DBs. Meanwhile, all the recordings of distorted data in Aurora-4 and Aurora-5 DBs are performed artificially. From this point, CHiME-4 DB can be supplementary to the artificial recording issue. Originated from the popular ASR workshop (CHiME challenge), CHiME-4 DB consists of both real and simulated recordings with additive noise and reverberation.

2.2.1 Aurora-4 DB

Aurora-4 DB [16] was made using 5k-word vocabulary based on the Wall Street Journal (WSJ) DB. The WSJ data were recorded with a primary Sennheiser microphone and with a secondary microphone in parallel. The recordings with the secondary microphone are used for enabling recognition experiments with different frequency characteristics in the transmission channel. An additional filtering is applied to consider the realistic frequency characteristics of terminals and equipment in the telecommunication area. Two standard frequency characteristics are used which have been defined by the ITU. The abbreviations G.712 and P.341 have been introduced as reference to these filters. The G.712 characteristic is defined for the frequency range of the usual telephone bandwidth up to 4 kHz and has a flat characteristic in the range between 300 and 3400 Hz. P.341 is defined for the frequency range up to 8 kHz and represents a band pass filter with a very low cut off frequency at the lower end and a cut off frequency at about 7 kHz at the higher end of the bandpass. These two filters can be applied to data sampled at 8 or 16 kHz, respectively. We use the 16 kHz sampled data.

The corpus has two training sets: clean- and multi-condition. Both clean- and multi-condition sets consist of the same 7138 utterances from 83 speakers. The clean-condition set consists of only the primary Sennheiser microphone data. One half of the utterances in the multi-condition set were recorded by the primary Sennheiser microphone and the other half were recorded using one of 18 different secondary microphones. Both halves include a combination of clean speech and speech corrupted by one of six different types of noises (car, babble, restaurant, street, airport and train station) at a range of signal-to-noise ratios (SNRs) between 10 and 20

Table 2.1: Aurora-4 DB (m: male, f: female).

	Training data	Development data	Evaluation data
Hour	15.1471	8.9694	9.4026
Utterance	7138	4620	4620
Speaker	83 (m: 42, f: 41)	10 (m: 6, f: 4)	8 (m: 5, f: 3)

dB. These noises represent realistic scenarios of application environments for mobile telephones. Some noises are fairly stationary like e.g. the car noise. Others contain non-stationary segments like e.g. the recordings on the street and at the airport. The SNR was defined as the ratio of signal to noise energy after filtering both speech and noise signals with P.341 filter characteristic.

The evaluation was conducted on the test set consisting of 330 utterances from 8 speakers. This test set was recorded by the primary microphone and a number of secondary microphones. These two sets were then each corrupted by the same six noises used in the training set at SNRs between 5 and 15 dB, creating a total of 14 test sets. These 14 sets were then grouped into 4 subsets based on the type of distortions: none (clean speech), additive noise only, channel distortion only and noise + channel distortion. For convenience, we denote these subsets by Set_A, Set_B, Set_C and Set_D, respectively. Note that the types of noises are common across training and test sets but the SNRs of the data are not.

For the validation test, we used the development set in Aurora-4 DB consisting of 330 utterances from 10 speakers not included in the training and test set speakers. A total of 14 sets with the same conditions as the test set were constructed. More detail information for Aurora-4 DB is given in Table 2.1.

2.2.2 Aurora-5 DB

Aurora-5 DB was developed to investigate the influence on the performance of ASR for a hands-free speech input in noisy room environments [13]. In Aurora-5, two test conditions are also included to study the influence of transmitting the speech in a mobile communication system. The number of test utterances was 8700 for each test condition.

In the Aurora-5, the test data consisted of two sets: G. 712 filtered and non-filtered sets summarized in Tables 2.2 and 2.3. The G. 712 filtered set comprised clean speech utterances to which randomly selected car or public space noise samples were added at SNR levels 0 to 15 dB. A car noise segment was randomly selected from 8 recordings that were made in two different cars under different conditions. As noise at public places a segment was randomly selected from 4 recordings at an airport, at a train station, inside a train and on the street. The GSM radio channel is also applied to simulate an influence for transmitting the noisy speech over a cellular telephone network. For the simulation of the GSM transmission, AMR speech codec was applied with various modes of bitrates and carrier-to-interference levels. The non-filtered set consisted of clean speech utterances to which randomly selected interior noises were added at SNR levels from 0 to 15 dB. The interior noise samples were recorded at a shopping mall, a restaurant, an exhibition hall, an office and a hotel lobby. Furthermore, to simulate the hands-free speech in a room, the clean speech signals are convoluted with the impulse responses of three different acoustic scenarios: hands-free in car (HFC), hands-free in office (HFO) and hands-free in living room (HFL). For this simulation, the reverberation times for the office and living rooms were randomly varied inside ranges of 0.3-0.4 and 0.4-0.5 seconds,

Table 2.2: G. 712 filtered test data set

Noise	Car Noise			Street Noise
		Hands-free in Car (HFC)	HFC & GSM (HFC-GSM)	GSM
SNR	Clean	Clean	Clean	Clean
	15	15	15	15
	10	10	10	10
	5	5	5	5
	0	0	0	0

Table 2.3: Non-filtered test data set

Noise	Interior Noise		
		Hands-free in Office (HFO)	Hands-free in Living Room (HFL)
SNR	Clean	Clean	Clean
	15	15	15
	10	10	10
	5	5	5
	0	0	0

respectively.

2.2.3 MC-WSJ-AV DB

MC-WSJ-AV corpus [14] can be categorized into three scenarios: single speaker stationary, single speaker moving and overlapping speakers scenarios. Since we are dealing with only the audio data in the single speaker stationary scenario, this section overviews the recording of the single speaker stationary scenario in MC-WSJ-AV database.

For the recording of the single speaker stationary scenario data, the data is recorded in three sites: The centre for speech technology research, edinburgh (UEDIN), The IDIAP research institute, Switzerland (IDIAP) and TNO Human Factors, the Netherlands (TNO). Instrumented meeting rooms installed at the three sites allow the audio to be fully synchronized. The layout of the UEDIN room with the positions of the microphone arrays and the six reading positions, is shown in Fig. 2.2. The room contains two eight-element circular microphone arrays, one mounted at the center and one at the end of the meeting room table. Array microphones are numbered 1-16. Cameras are mounted under Array 1 to give closeup views of participants in the seated locations. The six reading locations are indicated as Seat 1-4, Presentation and Whiteboard.

In addition, the speakers are provided with close-talking radio headset and lapel microphones. The TNO and IDIAP rooms contain the similar recording equipments, but differ in their physical layout and acoustic conditions. In the single speaker stationary condition, the speaker was asked to read sentences from six positions within the meeting room: four seated around the table, one standing at the whiteboard and one standing at the presentation screen. For each speaker, one sixth of the sentences.

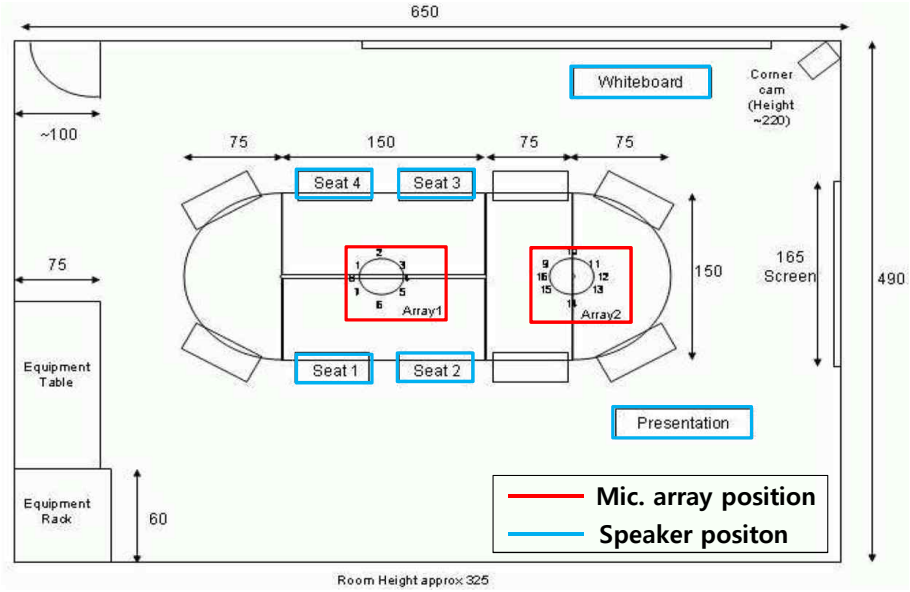


Figure 2.2: The layout of the UEDIN Instrumented Meeting Room.

2.2.4 CHiME-4 DB

The CHiME-4 [15] speech recordings were made using a 6-channel microphone array constructed by embedding omnidirectional microphones around the edge of a frame designed to hold a tablet computer. The array was designed to be held in landscape orientation with three microphones positioned along the top and bottom edges as indicated in 2.3. All microphones are forward facing except for channel 2 (shaded gray) which faces backwards and is flush with the rear of the 1 cm thick frame.

The microphone signals were recorded sample-synchronously using a 6-channel digital recorder. All recordings were made with 16 bits at 48 kHz and later downsampled to 16 kHz. Speech was recorded for training, development and test sets. Four native US talkers were recruited for each set (two male and two female). Speakers were instructed to read sentences that were presented on the tablet PC while holding

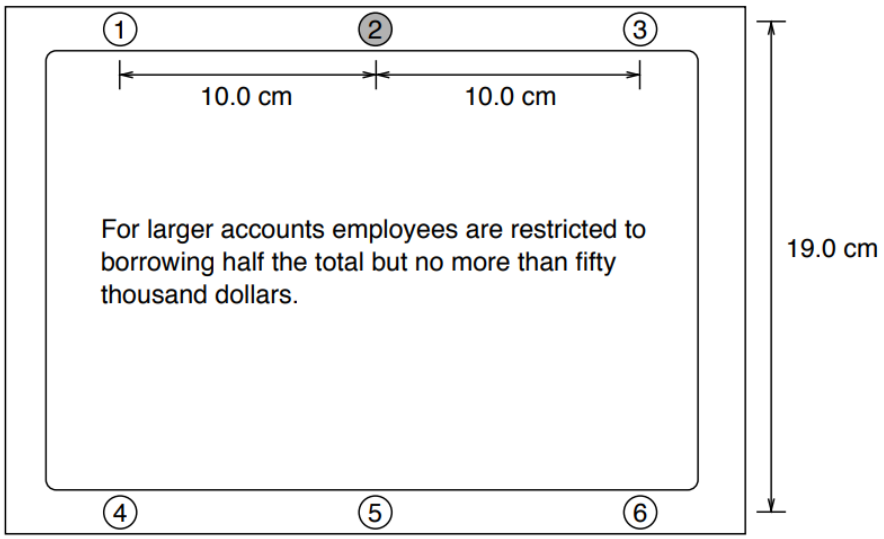


Figure 2.3: The geometry of the 6-channel CHiME-4 microphone array.

the device in any way that felt natural. Each speaker recorded utterances first in an IAC single-walled acoustically isolated booth and then in each of the following environments: on a bus (BUS), on a street junction (STR), in a cafe (CAF) and in a pedestrian area (PED).

The task was based on the WSJ0 5K ASR task. For the training data, 100 utterances were recorded by each speaker in each environment, totalling 1600 utterances selected at random from the full 7138 WSJ0 SI-84 training set. Speakers assigned to the 409 utterance development set or the 330 utterance final test set each spoke a 1/4 of each set in each environment resulting in 1636 (4×409) and 1320 (4×330) utterances for development and final testing respectively.

Chapter 3

Two-stage Noise-aware Training for Environment-robust Speech Recognition

3.1 Introduction

Ever since the deep neural network (DNN)-based acoustic model appeared, the recognition performance of automatic speech recognition (ASR) has been greatly improved [1], [2], [17], [18]. Based on this achievement, researches on DNN-based techniques for noise robustness are also in progress. Among various approaches, adaptation technique employing auxiliary features with acoustic context information demonstrated their potential.

One of the simplest methods of these approaches is to augment the auxiliary features with the input vector of the network. As an example, the technique referred to as noise-aware training (NAT) attained state-of-the-art results on Aurora-4 task [10].

NAT enables the DNN to learn the relationship among noisy input, noise features and target vectors corresponding to the phonetic identity by augmenting an estimate of the noise present in the input signal. Due to its simple implementation and good performance, NAT has already been applied actively in speech enhancement and robust ASR.

Despite its success in robust ASR, we cannot be certain whether NAT is an optimal method in taking advantage of the inherent robustness of the DNN framework. Although NAT somewhat contributes to the noise robustness of DNN, its performance in adverse environment is still far from that shown in clean condition. One of the fundamental reasons for this phenomenon is that the current NAT framework is considered insufficient to make the DNN implement the mapping from noisy speech and noise estimates to phonetic targets as clearly as it addresses the relationship between clean speech and the corresponding phonetic targets. A promising way to improve NAT may be to extract some representation relevant to clean speech features and then to implement the mapping from this representation to the phonetic targets.

In this chapter, we propose a novel approach to DNN training which can be a solution to the aforementioned issue of NAT. The main idea of the proposed approach is to let the DNN clarify the relationship among noisy features, noise estimates and phonetic targets only after reconstructing the clean features. In order to accomplish this, the proposed technique cascades two individually fine-tuned DNNs into a single DNN. The first DNN performs reconstruction of the clean features from noisy features when noise estimates are augmented. Then the next DNN attempts to learn the mapping between the reconstructed features and the phonetic targets. The performance of the proposed approach is evaluated on the Aurora-5 task and also in

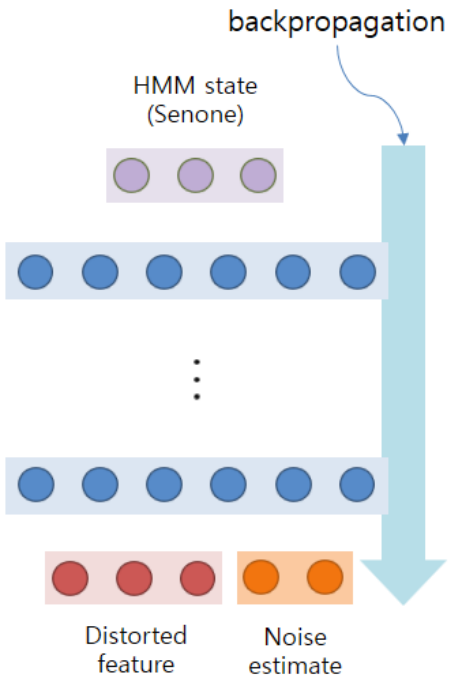


Figure 3.1: DNN structure of noise-aware training.

some mismatched noise conditions, and better performance is observed compared to the conventional NAT.

3.2 Noise-aware Training

The structure of NAT is represented in Fig. 3.1. For a simple problem formulation, we consider acoustic environments where the background noises are dominant factors of speech degradation. Let us denote an observed noisy feature, the corresponding unknown clean feature, the corrupting noise and a HMM state identity being extracted at the t-th frame as $\mathbf{y}_t$, $\mathbf{x}_t$, $\mathbf{n}_t$ and $\mathbf{s}_t$, respectively. Additionally, we denote a subsequence of vectors $\mathbf{x}_{m_1}\mathbf{x}_{m_1+1}\cdots\mathbf{x}_{m_2}$ from frame index m_1 to m_2 as $\mathbf{x}_{m_1:m_2}$. Under the general framework of HMM-based recognition, we assume that

there exists an unknown underlying function that approximates the posterior probabilities of the HMM states given as follows:

$$p(\mathbf{s}_t|\mathbf{y}_t) \cong f(\mathbf{y}_{t-\tau:t+\tau}, \mathbf{n}_{t-\tau:t+\tau}) \tag{3.1}$$

where $f(\cdot)$ represents the function that maps the noisy and noise features to the corresponding HMM state identity which contains phonetic information and the subscript τ represents the temporal coverage which is required for figuring out the contextual information of the speech signal.

Since the true noise features $\mathbf{n}_{t-\tau:t+\tau}$ in (3.1) are unknown, NAT replaces them with a single noise estimate. The input vector of NAT is formed by augmenting the noise estimate with a window of consecutive frames of noisy feature, i.e.,

$$\mathbf{v}_t = [\mathbf{y}_{t-\tau:t+\tau}, \widehat{\mathbf{n}}_t] \tag{3.2}$$

where a window of $2\tau + 1$ frames of noisy speech features and $\widehat{\mathbf{n}}_t$ represents a noise estimate. The target vector of the NAT network is given as the one-hot encoding label concerned with the tied HMM states (senone) like common DNN-based acoustic models. By applying this simple process to both training and decoding, the DNN can automatically learn the complex mapping from the noisy speech and noise estimate to the HMM state labels.

However, even though this approach guarantees a certain level of improvement in noise robustness, we need to check whether the non-linear mapping obtained from NAT can be generalized well. Although NAT aims to generate internal representations that are robust to noise, when comparing its recognition performance in noisy environment with that in clean environment, we can easily discover that there still exists a large performance gap. For this reason, we need a more sophisticated technique to improve the modeling power of the NAT.

3.3 Two-stage NAT

In this section, we propose a novel approach to improve NAT. The basic idea of the proposed approach starts from the assumption that the underlying function $f(\cdot)$ in (3.1) can be expresses as a composition of two separate functions as follows:

$$p(\mathbf{s}_t|\mathbf{y}_t) \cong f(\mathbf{y}_{t-\tau:t+\tau}, \mathbf{n}_{t-\tau:t+\tau}) \cong h \circ g(\mathbf{y}_{t-\tau:t+\tau}, \mathbf{n}_{t-\tau:t+\tau}) \tag{3.3}$$

where the output of $g(\cdot)$ is a clean feature vector stream,

$$\mathbf{x}_{t-\tau:t+\tau} \cong g(\mathbf{y}_{t-\tau:t+\tau}, \mathbf{n}_{t-\tau:t+\tau}), \tag{3.4}$$

and

$$p(\mathbf{s}_t|\mathbf{y}_t) \cong h(\mathbf{x}_{t-\tau:t+\tau}). \tag{3.5}$$

In (3.3)-(3.5), $g(\cdot)$ represents a function dealing with the mapping from the noisy and noise features to the clean speech features and $h(\cdot)$ is a function predicting the phonetic target based on the clean speech feature stream. To mimic this function structure, we propose a DNN as shown in Fig. 3.2. The whole DNN is constructed by concatenating two individually fine-tuned DNNs and each separate DNN approximates the function $g(\cdot)$ and $h(\cdot)$ in (3.3). The first DNN is applied to separate the clean speech features from the corruption noises. We call this DNN the lower DNN since it is placed in the lower part of the DNN in Fig. 3.2. The second DNN which is called the upper DNN, deals with modeling the relationship between the output vector generated by the lower DNN and the phonetic target.

3.3.1 Lower DNN

The output layer of the lower DNN corresponds to the clean speech features and the noise features and the input layer is given by (3.2). The output vector of the

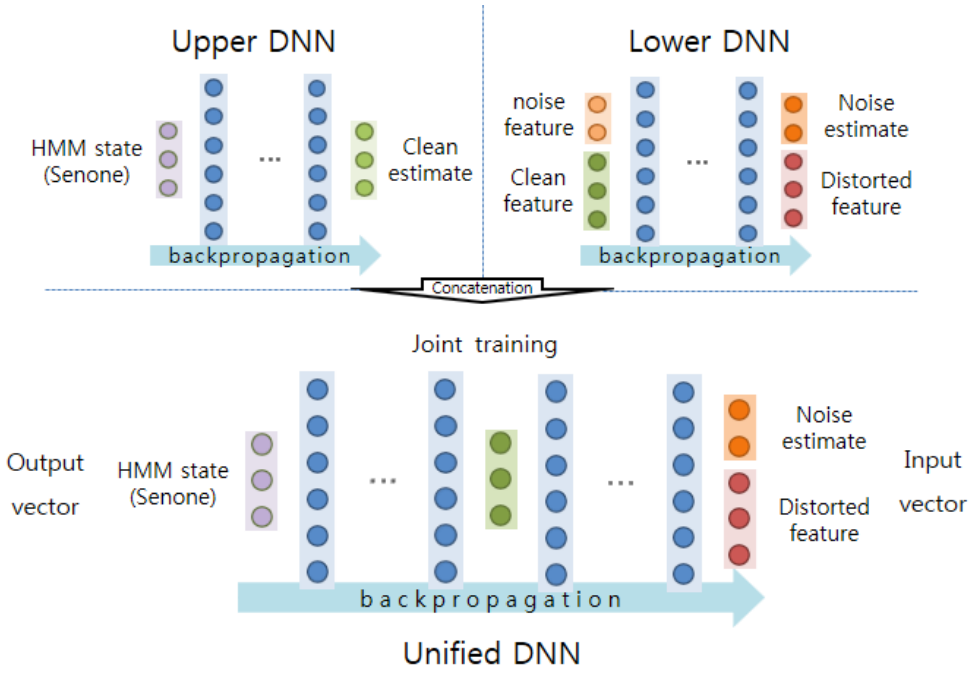


Figure 3.2: DNN structure of proposed technique.

lower DNN can be represented as follows:

$$\widehat{\mathbf{v}}_t = [\widehat{\mathbf{x}}_{t-\tau:t+\tau}] \tag{3.6}$$

where a window of $2\tau + 1$ frames of clean speech feature estimates. To obtain the noise estimate $\widehat{\mathbf{n}}_t$ in (2), a time-varying environmental estimation approach based on interacting multiple model (IMM) algorithm is utilized. By reflecting the dynamic environmental information estimated from the IMM technique to the input of the network at each frame, we can expect the lower DNN to reconstruct clean features irrespective of environmental conditions.

Meanwhile, insufficient information about the true noise makes the lower DNN distort reconstructed clean features and this naturally leads to improper mapping between the input and phonetic target. To compensate for this problem, we additionally apply multi-task learning (MTL). In a general MTL framework, multi-task

objective function J_{MTL} is expressed as follows:

$$J_{MTL} = J + \alpha J_{aux} \tag{3.7}$$

where J and J_{aux} denote the objective functions of primary and secondary tasks respectively, and α is the weight parameter which determines how much importance the secondary task has. After the training is over, only the primary task is performed and the parameters associated with the output of the secondary task are discarded.

In the lower network training, MTL is applied to the lower DNN with true noise feature. Specifically, the target vector of the lower DNN adds noise feature corresponding to noise estimate feature of the input vector. Therefore, the objective function of the extended lower DNN J_L can be represented as follows:

$$J_L = \sum_t ||\mathbf{o}_t - \widehat{\mathbf{o}}_t||^2 + \alpha \sum_t ||\mathbf{n}_t - \widehat{\mathbf{n}}_t||^2 \tag{3.8}$$

where $\mathbf{o}_t$ and $\widehat{\mathbf{o}}_t$ denote the target and output vectors of the lower DNN. By flowing back the information of the true noise feature, the extended lower DNN can absorb the environmental information more distinctly. Particularly, the shared structure serves to improve the generalization of the model and its accuracy on an unseen test set. In this technique, α was set to 1.

3.3.2 Upper DNN

In the stage of upper DNN training, the network learns the mapping between the output vector of the lower DNN $\widehat{\mathbf{v}}_t$ in (3.6) and the corresponding one-hot encoding label which contains information of the HMM states. Through the mapping, the prediction of the posterior probabilities of the HMM states from the reconstructed features can be enacted. Since $\widehat{\mathbf{v}}_t$ is acquired by the lower DNN, the reconstructed

vector is free from information loss caused by using linear approximations which are used in the conventional techniques.

3.3.3 Joint Training

After the training of the upper DNN is over, two different networks are cascaded to form a single larger DNN and the unified DNN jointly adjusts the weights using the backpropagation algorithm. In detail, the error signal between the phonetic target and the output of the unified DNN flows back to the clean estimate feature layer and the extended lower DNN, consequently training all the parameters. With this series of processes, learning the relationship among the noisy, noise estimate, true noise features and phonetic target labels can be enhanced by guiding the DNN through the intermediate level features.

3.4 Experiments

To evaluate the speech recognition performance of the proposed approach, we performed a series of experiments in both matched and mismatched noise conditions. While the matched noise conditions were obtained from Aurora-5 task where the detailed information is given in 2.2.2.The mismatched noise conditions were made using 100 non-speech environmental sounds.

3.4.1 GMM-HMM System

In these experiments, we used multi-condition training data for construction of all the DNN-based acoustic models. In order to create phonetic labels of the training data, the GMM-HMM systems were built based on the clean speech data provided by

the G. 712 filtered and non-filtered data sets which is counterpart of multi-condition training data. These systems consisted of 179 HMMs states and 4 Gaussians per state trained using maximum likelihood estimation. The number of utterances used for HMM training was 8623 for each data set. The input features were 39-dimensional MFCC features (static plus first and second order delta features) and cepstral mean normalization was performed. The training of the HMM parameters and Viterbi decoding for speech recognition was carried out using HTK [19].

3.4.2 Training and Structures of DNN-based Techniques

The performance of the proposed method was compared with three different versions of DNN-based approaches. The compared techniques are

- *Baseline*: Basic multi-condition DNN-HMM,
- *NAT*: Noise-aware training [10],
- *Proposed*: Two-stage noise-aware training

For training all the DNN-based acoustic models, LMFB feature of 23-dimension was used. As in the case of MFCC feature above, both the first and second-order derivative of LMFB features were used.

The input layer for *Baseline* was formed from a context window of 11 frames having 759 visible units for the network and that of *NAT* had total 828 visible units by augmenting the input vector of *NAT* with the IMM-based noise estimate. Both DNNs had 11 hidden layers with 2048 ReLUs in each layer and the final soft-max output layer had 179 units, each corresponding to the states of the HMM systems. The fine-tuning of the two networks were performed using cross entropy as the loss function by error back propagation supervised by senones for frames.

The lower DNN had hidden five layers in total and the number of nodes in each hidden layer was set to be 2048 ReLUs. The input layer of the lower DNN was equal to that of *NAT*.

The upper DNN had 5 hidden layers with 2048 ReLUs. And the final soft-max output layer had 179 units in common with the other DNN-HMMs above. The rest of the training configurations were the same with those of the other DNN-HMMs. The parameters of the DNN-based techniques were randomly initialized and fine-tuned using SGD algorithm.

Mini-batch size for the SGD algorithm was set to be 256 for all of the DNN-based techniques. The momentum was set to be 0.5 at the first epoch and increased to 0.9 afterward. The learning rate was initially set to be 0.01 and exponentially decayed over each epoch with a decaying factor of 0.9 except for the cases of two lower DNNs and joint training of the proposed method. For two lower DNNs and the joint training, learning rate was initially set to be 0.0005 and exponentially decayed over each epoch with a decaying factor of 0.95. All the training of DNN-based techniques were stopped after 50 epochs.

All the techniques evaluated in this experiments were based on wide and very deep DNN structures. To prevent overfitting, dropout was also applied [20]. The retention rate of dropout was 0.8.

3.4.3 Performance Evaluation

Table 4.1 shows the results of the various DNN-based techniques. We can see that the proposed method outperformed other DNN-based techniques irrespective of the SNRs. Further improvement was observed when the dropout training was applied. The average relative error rate reductions (RERRs) of *Proposed* over *NAT*

Table 3.1: WERs (%) on Aurora-5 task according to variety of DNN-based acoustic models

SNR (dB)	Non-filtered			G.712 filtered		
Method	*Baseline*	*NAT*	*Proposed*	*Baseline*	*NAT*	*Proposed*
Clean	1.38	1.28	**0.89**	0.95	0.78	**0.70**
15	1.85	1.87	**1.28**	1.32	1.18	**0.82**
10	3.21	3.14	**2.35**	2.18	1.87	**1.37**
5	7.67	7.55	**6.23**	4.65	4.35	**3.52**
0	20.55	20.01	**18.87**	12.91	12.25	**11.29**
Average	6.93	6.77	**5.92**	4.40	4.09	**3.54**

Table 3.2: WERs (%) on the noise-mismatched test set according to variety of DNN-based acoustic models

SNR (dB)	Non-filtered			G.712 filtered		
Method	*Baseline*	*NAT*	*Proposed*	*Baseline*	*NAT*	*Proposed*
Clean	1.38	1.28	**0.89**	0.95	0.78	**0.70**
15	3.68	3.12	**2.62**	4.39	4.29	**4.05**
10	9.42	5.88	**5.01**	10.28	10.35	**7.89**
5	23.78	12.11	**10.56**	22.58	18.12	**14.89**
0	44.25	24.02	**19.76**	41.52	29.75	**16.78**
Average	16.50	9.28	**7.77**	15.94	12.66	**10.86**

were 12.5% and 13.36% in non-filtered and G.712 filtered set.

To evaluate the proposed technique in training-test mismatched noise condi-

Table 3.3: Computation complexity measurement of variety of DNN-based acoustic models

Method	*Baseline*	*NAT*	*Proposed*
No. of param.	43.9 M	44.0 M	38.7 M
xRT	0.025	0.125	0.122

tions, we constructed the noise-mismatched test sets by mixing the clean speech of non-filtered and G. 712 filtered sets with four noises included in 100 non-speech environmental sounds [21]. Four types of noise were chosen from 100 noise types : animal, water, wind sound and phone dialing. Each noise types were added to the G. 712 filtered and non-filtered sets at SNRs between 0 and 15 dB with equal rate. From the results in Table 4.4, we can see that the proposed technique is more effective in mismatched noise conditions. Especially, when dropout training is performed the average relative error rate reductions (RERRs) of *Proposed* over *NAT* were 16.31% and 14.19% in noise-mismatched non-filtered and G.712 filtered set.

3.5 Summary

In this chapter, we have proposed a novel technique of DNN-based acoustic model designed for effective usage of multi-condition data and its noise estimate. The proposed technique addressed the mapping from noisy speech and noise estimates to phonetic targets effectively by concatenating two fine-tuned DNNs and training the unified network jointly. Through a series of experiments on Aurora-5 task and mismatched noise conditions, we have found that the proposed technique outperforms NAT in word accuracy on both matched and mismatched conditions.

Chapter 4

DNN-based Feature Enhancement for Robust Multichannel Speech Recognition

4.1 Introduction

Since the introduction of deep neural network (DNN)-based acoustic model to automatic speech recognition (ASR), various studies on DNN-based techniques for robust ASR have been in progress. Due to the progresses above, the ASR system has achieved great performance in close-talking environments. However, recent developments in speech and audio applications such as hearing aids and hands-free speech communication systems require speech acquisition in distant-talking environments.

Unfortunately, as the distance from the speaker and the microphone increases, the recorded speech becomes more distorted due to the background noise and room reverberation. Although it may be possible to acquire the speech in close-talking environments by using a headset microphone, it is not a general solution because of the inefficiency in terms of cost and ease of use. Consequently, ASR performance in distant-talking environments is still far from that shown in close-talking environments.

In order to overcome this difficulty, various researches have focused on techniques for efficiently integrating the information obtained from multiple distant microphones to improve the ASR performance. One of the most conventional multichannel-based techniques is the beamformer method, which enhances the signals emanating from a particular location by individual microphone arrays. The simplest technique is the delay-and-sum (DS) beamformer [22], which compensates the delays of the microphone inputs so that only the target signal from a particular direction synchronizes with. In addition, there are many sophisticated beamforming methods [23], [24] which optimize the beamformers to produce a spatial pattern with a dominant response for the location of interest.

Feature mapping techniques based on DNN have been also investigated recently. DNN-based feature enhancement techniques [3], [4] have already been widely employed in robust ASR due to their advantage in directly representing the arbitrary unknown mapping between the noisy and clean features unlike the conventional techniques [25]–[28] which usually require specific assumptions or formulations. Especially, [4] showed that the feature mapping technique combining beamformer and DNN improves the performance of the ASR system in multichannel distant speech recognition.

Meanwhile, recent researches on joint training technique of DNN [8], [9] have drawn attention. builds a DNN by concatenating two independently trained DNNs and jointly adjusting the parameters. Through this training technique, the synergy between two DNNs can be amplified. Traditionally, this joint training framework has been applied to incorporate two different tasks into one universal task, i.e., integrating speech separation and acoustic modeling [9]. In addition to the usage above, the joint training technique can be used for training a DNN in charge of a single task elaborately. In these circumstances, the performance of DNN depends on deciding which types of features are represented in the intermediate layer where junction between two DNNs occur. In [29], a performance of DNN was enhanced by giving appropriate intermediate concepts which the DNN should represent in the mid-level.

In this chapter, we propose a novel DNN-based feature enhancement technique for multichannel distant speech recognition in modern multichannel environments where various types of microphone data are given as training data. The main contribution of the proposed approach is to construct a multichannel-based feature mapping DNN algorithm by properly combining a conventional beamformer, DNN and its joint training technique with lapel microphone data which has an intermediate level of acoustic information between DNN input and the target. To implement the technique making use of various microphone types and evaluate the performance, we used a data set of single speaker scenario from MC-WSJ-AV corpus [14] which is a re-recorded version of WSJCAM0 [30] in a meeting room environment. More detail information for MC-WSJ-AV corpus is given in 2.2.3.

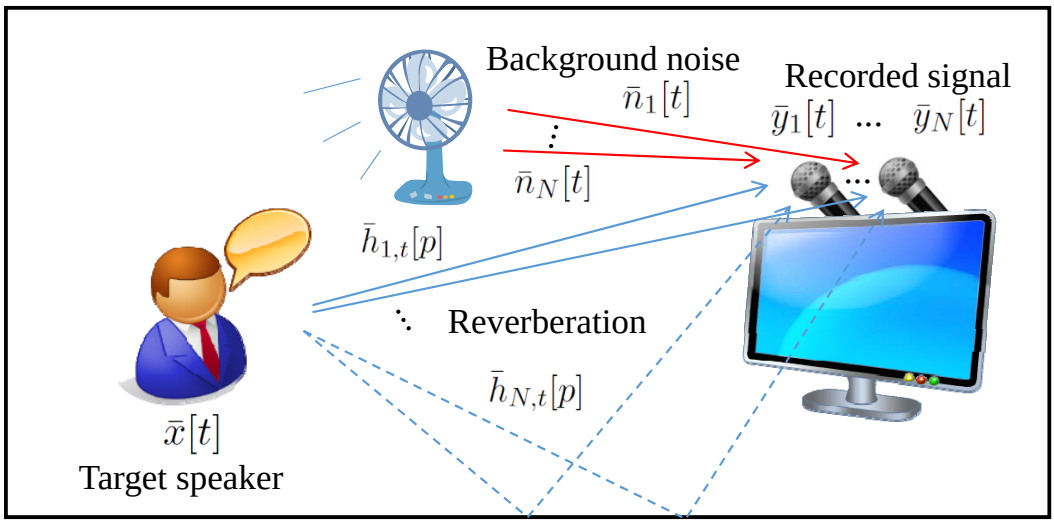


Figure 4.1: Reverberant noisy environment in multi-channel scenario.

4.2 Observation Model in Multi-Channel Reverberant Noisy Environment

We consider a typical hands-free scenario for ASR in which multiple microphones are used as shown in Fig. 4.1. The target speaker is located in a certain distance from the microphones in an enclosed room, which results in acoustic reverberation. Let $\bar{y}_i[t]$ be the signal obtained from the i-th microphone with $t \in \{0, 1, \cdots\}$ denoting the time index. If $\bar{x}[t]$ is the target speech signal and $\bar{h}_{i,t}[p]$ represents the RIR from the target speaker to the i-th microphone with corresponding tap index $p \in \{0, 1, \cdots\}$, then

$$\bar{y}_i[t] = \sum_{p=0}^{\infty} \bar{h}_{i,t}[p]\bar{x}[t-p] + \bar{n}_i[t] \tag{4.1}$$

where $\bar{n}_i[t]$ is the background noise added to the i-th microphone input.

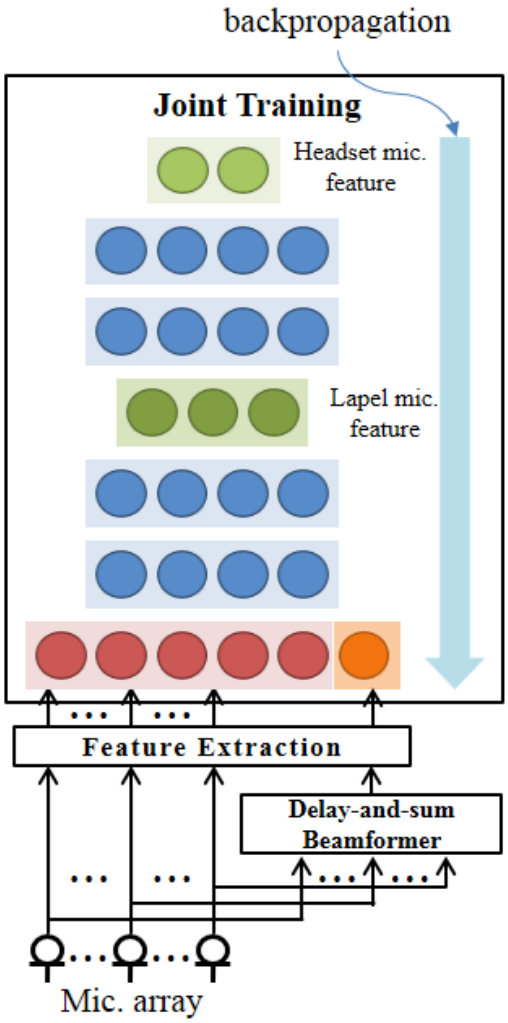


Figure 4.2: The schematic diagram of proposed technique.

4.3 Proposed Approach

In this chapter, the (m)-th array microphone feature, the DS-beamformed feature from the array, lapel microphone feature and headset microphone feature being extracted at the t-th frame are denoted as $\mathbf{a}_t^{(m)}$, $\mathbf{b}_t$, $\mathbf{l}_t$ and $\mathbf{h}_t$, respectively.

We propose a novel DNN-based feature enhancement approach for multichannel

distant speech recognition. The purpose of our technique is to estimate the clean features from the distant array features. However, there exists two problems for enabling the DNN to achieve this adverse task. The first problem is the phase differences among each signal of array microphones originated from the distances between the speaker and each microphone. And the second problem, which is more serious, is the lack of acoustic information of the array. Due to the distances between each of the array microphones and the speaker, the microphones have low ratio of direct-to-reverberant speech energy which becomes a huge limitation on reconstructing the clean speech entirely. To compensate for these problems, we propose the DNN as shown in Fig. 4.2.

The proposed DNN is constructed by concatenating two individually fine-tuned DNNs and training the unified DNN jointly. We call the first DNN as lower DNN since it is placed in the lower part of the DNN in Fig. 4.2. The second DNN which is called the upper DNN, deals with modeling the relationship between the output vector generated by the lower DNN and the headset microphone feature.

4.3.1 Lower DNN

For training the lower DNN, DS beamforming [22] is employed to the microphone array to align the phases of microphone inputs. Once the beamforming has been applied, the input vector of the lower DNN $\mathbf{v}_t$ is formed by concatenating a window of several adjacent frames of feature from the beamformed source and additional windows covering each array microphone features, i.e.,

$$\mathbf{v}_t = [\mathbf{a}^{(1)}_{t-\tau:t+\tau}, \mathbf{a}^{(2)}_{t-\tau:t+\tau}, \cdots, \mathbf{a}^{(M-1)}_{t-\tau:t+\tau}, \mathbf{a}^{(M)}_{t-\tau:t+\tau}, \mathbf{b}_{t-\tau:t+\tau}] \tag{4.2}$$

where τ represents the temporal coverage required for figuring out the clean feature of t-th frame and M represents the number of the array elements. This input structure helps the lower DNN to learn the correlations among features of array microphones. As the target vector of the network, we used a window of several frames of feature obtained from lapel microphone which has a much higher ratio of direct-to-reverberant speech energy than those of the array microphones but lower than those of the headset microphones. Therefore, the lower DNN output can be represented as follows:

$$\widehat{\mathbf{o}}_t^L = [\ \widehat{\mathbf{l}}_{t-\tau:t+\tau}]. \tag{4.3}$$

4.3.2 Upper DNN and Joint Training

In the training stage of the upper DNN training, the network learns the mapping between the output vector of the lower DNN and the corresponding headset microphone feature which can be interpreted as a ideal clean feature. The mapping can be represented as follows:

$$\widehat{\mathbf{o}}_t^U = [\widehat{\mathbf{h}}_t] \cong f(\widehat{\mathbf{l}}_{t-\tau:t+\tau}). \tag{4.4}$$

Here, function f is a function which deals with the mapping from the reconstructed lapel microphone features to the headset microphone feature. Since the clean features are estimated from the reconstructed lapel features which have more abundant acoustic information than the array features, we can expect more accurate reconstruction of clean features.

After training the upper DNN, two different networks are cascaded to form a single larger DNN and the unified DNN jointly adjusts the weights using the back-propagation algorithm. In detail, the error signal between the clean target and the

output of unified DNN flows back to the lapel microphone feature layer and the lower DNN, and consequently training all the parameters. With this series of processes, learning the relationship between the array features and the headset features can be enhanced by guiding the DNN through the intermediate level features. For training all the DNNs in the proposed method, the SGD algorithm is used to minimize the mean squared error (MSE) function which is given by

$$C_{MSE} = \frac{1}{T}\sum_{t=1}^{T} ||\mathbf{O}_t - \widehat{\mathbf{O}}_t||^2 \tag{4.5}$$

where $\mathbf{O}_t$, $\widehat{\mathbf{O}}_t$, and T denote the target, output vector of network and number of training samples, respectively.

4.4 Experiments

The proposed technique was trained on development set (DEV) and its performance was evaluated on evaluation set (EVAL1) of MC-WSJ-AV DB. The selection of read sentences for these sets was based on the development and evaluation sets of the WSJCAM0 British English corpus [30]. Each speaker prompt contained 17 adaptation sentences, 40 sentences from the 5000-word sub-corpus, respectively.

In this section, some basic experimental results obtained from DS-beamformed source (*DS*) of microphone array, headset microphone (*Headset*), lapel microphone (*Lapel*) and single distant microphone (*SDM*) recordings were presented. Here, the microphone array refers to Array 1 which is the left one among the two arrays in Figure 1 and single distant microphone is the no. 1 microphone of the Array 1. Also, the comparison of performances with conventional DNN-based feature mapping methods were included.

4.4.1 Recognition System and Feature Extraction

A baseline DNN-HMM system was trained on the WSJCAM0 database. The training set consisted of 53 male and 39 female speakers. We used the Kaldi speech recognition toolkit [31] for feature extraction, acoustic modeling of ASR and ASR decoding. For feature extraction, 13-dimensional MFCCs (including C_0) with their first and second derivatives were extracted and the cepstral mean normalization algorithm was applied for each speaker. In order to provide the target alignment information for the DNN-based acoustic model, we built a GMM-HMM system with 2047 senones and 15026 Gaussian mixtures in total. The target senone labels of the DNN-HMM system were obtained over the training data. As for the language model, we applied the standard 5k WSJ trigram language models.

For the DNN training of the acoustic model, we applied five hidden layers with 2048 nodes. As for the input of the DNNs, input features consisted of consecutive 11-frame (5 frames on each side of the current frame) context window of 13 dimensional MFCC features with their first and second order derivatives, resulting with the input dimension of 429. The input features of the DNNs were normalized to have zero mean and unit variance. The output dimension of the DNN was 2047. Generative pre-training algorithm for the restricted Boltzmann machines was carried out to initialize the DNN parameters as described in [32]. The errors between the DNN output and the target senone labels were calculated according to the cross-entropy criterion [2]. In order to speed up the training, we applied the learning rate scheduling scheme and the stop criteria presented in [32].

4.4.2 Training and Structures of DNN-based Techniques

The performance of the proposed method was compared with different versions of DNN-based feature enhancement approaches. The compared techniques are

- *FE-SDM*: mapping single array microphone into a clean target source,
- *FE-DS*: mapping DS-beamformed source of the array into a clean target source,
- *FE-PMWF*: mapping adaptive beamformed source of the array into a clean target source,
- *FE-Array*: mapping multiple sources from microphone array into a clean target source,
- *FE-PMWF&Array*: mapping multiple sources including the sources from the microphone array and adaptive beamformed source of the array into a clean target source,
- *FE-DS&Array*: mapping multiple sources including the sources from the microphone array and DS-beamformed source of the array into a clean target source,
- *FE-Array-Joint*: mapping multiple sources from microphone array into a clean target source with applying the joint training framework via the lapel microphone feature,
- *FE-PMWF&Array-Joint*: mapping multiple sources including the sources from the microphone array and adaptive beamformed source of the array into a clean target source with applying the joint training framework via the lapel microphone feature.

In implementing DNN-based techniques using the adaptive beamforming, spectro-temporal parameterized multichannel non-causal Wiener filter-based enhancement technique (PMWF) was used. For training all the DNN-based feature enhancement techniques, we used cepstral mean normalized MFCC feature of 13 dimension with their first and second derivatives as an input. All the techniques used one or more windows depending on the number of sources and each window consists of 11 consecutive MFCCs. Meanwhile, the feature mapping DNNs commonly estimated 13-dimensional static MFCC of current frame and the outputs of DNNs were fed into the recognizer after extraction of their dynamic component. Table 4.4 shows the input and output dimensions of each DNN-based techniques. The networks had 5 hidden layers with 1024 ReLUs [33] are applied except for the proposed technique which contains the intermediate layer because of its unique structure. The parameters of the DNN-based techniques are randomly initialized and fine-tuned using SGD algorithm with minimum MSE objective function like those of the proposed method.

Mini-batch size for the SGD algorithm was set to be 256 for all of the DNN-based feature enhancement techniques. The momentum was set to be 0.5 at the first epoch and increased to 0.9 afterward. The learning rate was initially set to be 0.01 and exponentially decayed over each epoch with decaying factor of 0.9 except for the cases of the lower DNN and joint training of the proposed method. For lower DNN and the joint training, learning rate was initially set to be 0.001 and exponentially decayed over each epoch with a decaying factor of 0.95. All the training of DNN-based techniques were stopped after 50 epochs.

Table 4.1: WERs (%) on EVAL1 according to various source types

Channel	WER (%)
SDM	58.00
PMWF	46.14
DS	41.97
Lapel	13.18
Headset	7.49

4.4.3 Dropout

As one of the most well-known regularization techniques, dropout was also applied. Dropout is a method that improves the generalization ability of the DNN. It can be easily implemented by randomly dropping the input and hidden neuron units. As pointed out by Hinton et al. [34], dropout can be considered as a bagging technique that averages over a large amount of models with shared parameters of the DNN. A dropout percentage of 20% was applied to every DNN-based feature enhancement technique.

4.4.4 Performance Evaluation

Table 4.1 and Table 4.3 show the results according to various source types and DNN-based techniques, respectively. Comparison among the DNN-based approaches shows that high variety of input structure of the DNN guarantees better performance. We can see that the proposed method outperformed other DNN-based techniques including *FE-DS&Array* which has the same input structure but more

Table 4.2: Input and output dimensions of the DNN-based techniques.

Method	Input dim.	Output dim.
FE-SDM	429	13
FE-DS	429	13
FE-PMWF	429	13
FE-Array	3432	13
FE-PMWF&Array	3861	13
FE-DS&Array	3861	13
FE-Array-Joint	3432	13
FE-PMWF&Array-Joint	3861	13
Proposed	3861	13

parameters than the proposed approach. Meanwhile, when the techniques employing DS The average relative error rate reductions (RERRs) of the proposed method over *FE-DS&Array* was 9.8%. This confirms that our proposed approach which intervenes the DNN through information of reconstructed lapel microphone data can be effective in making the network to learn the complicated relationship between features from the distant microphone array, DS-beamformer and headset microphone sources.

Table 4.3: WERs (%) on EVAL1 according to variety of DNN-based feature enhancement techniques.

Method	WER (%)
FE-SDM	25.88
FE-DS	23.52
FE-PMWF	21.91
FE-Array	20.44
FE-PMWF&Array	20.11
FE-DS&Array	19.63
FE-Array-Joint	18.38
FE-PMWF&Array-Joint	18.06
Proposed	17.70

4.5 Summary

In this paper, we have proposed a novel DNN-based feature enhancement approach for multichannel distant speech recognition. The proposed approach constructed a multichannel-based feature mapping DNN using conventional beamformer, DNN and its joint training technique with lapel microphone data. Through a series of experiments on MC-WSJ-AV corpus, we have found that the proposed technique clarifies the relationship between the features obtained from distant microphone array and clean speech.

Table 4.4: Computation complexity measurement of the DNN-based techniques.

Method	No. of param.	xRT
FE-SDM	4.65 M	0.003
FE-DS	4.65 M	0.047
FE-PMWF	4.65 M	0.073
FE-Array	7.72 M	0.006
FE-PMWF&Array	8.61 M	0.051
FE-DS&Array	8.61 M	0.077
FE-Array-Joint	6.50 M	0.005
FE-PMWF&Array-Joint	6.94 M	0.049
Proposed	6.94 M	0.075

Chapter 5

Uncertainty-aware Training for DNN-HMM System using Variational Inference

5.1 Introduction

Although the DNN-based techniques introduced in previous chapters have shown better performances, there still exists the limitation of them., the estimated clean features or the phonetic targets may still be unreliable due to various sources of uncertainty. Yet, these sources of uncertainty are mostly overlooked when applying DNN-based techniques, which eventually tampers with model performance. When the test data contains unseen environmental effects (e.g., noise, reverberation, and speaker and channel mismatch) which are seldom observed in the training data, the accuracy of the estimator decreases and this degrades the overall performance of the ASR system.

Uncertainty decoding (UD) is a well-known approach in HMM-based ASR to addresses such issues effectively [35]–[40]. The main idea is to employ a stochastic process, instead of a deterministic one, in order to describe the mapping from the observed noisy features to the clean features. More specifically, UD, given a degraded input data, exploits a statistical model from which the posterior distributions of the unknown clean speech features are learned.

The key to the successful implementation of the UD technique is to determine how to model the posterior distribution based on which the marginalized likelihoods are computed. Amongst myriads of propositions, [41]–[50] attempt to reflect the input uncertainty in the feature domain with the assumption that uncertainty may be represented by specific statistical models (e.g., Gaussians and GMM). Especially, studies that additionally consider modified variances of each Gaussian component in the update of GMM-HMM parameters obtained remarkable performance [35]–[37], [51].

Inspired by prior work, efforts have been recently made to utilize deep learning techniques in the UD setting. For example, [52] and [53] implement Gaussian marginalization approximation approach to the conventional DNN-based inference. Despite their impressive performance, these neural network-based techniques cannot utilize softmax layers, which serve as the output layer, of the DNN-based acoustic model. This makes the techniques in [52] and [53] incompatible with the neural network-based acoustic model. On the other hand, [54]–[57] use numerical sampling in order to account for uncertainty in their DNN-HMM framework. However, these approaches process a single sample input by feeding in multiple samples, hence invoking inefficiency in terms of computational costs. Although [57] attempts to tackle this issue by the means of unscented transformation which considers only a rela-

tively reasonable number of samples during the training and decoding processes as compared to the other conventional sampling-based approach, the trade-off between performance and computational burden still remains to be an issue in the real world.

In this chapter, we propose a novel deep learning-based acoustic modeling technique which systematically measures and takes account of the uncertainty inherent in the input features using a single deep network. Our method distinguishes itself from the existing UD studies using NN-based acoustic model in two perspectives. Firstly, we divide the input uncertainty into two different domains: clean feature estimation and environment estimation. Secondly, instead of sampling, uncertainty information is fed into the NN-based acoustic model in the form of supplementary features as introduced in [7]–[12]. Such an approach allows our method to take account of input uncertainty in estimation with a relatively little increase in computational cost.

Our proposed technique, the uncertainty-aware training (UAT), namely, employs variational autoencoder (VAE) [58], [59], one of the widely used variational inference (VIF) techniques, which allows the extraction of robust features along with the associated uncertainties. VAE performs efficient inference under the assumption that the observed data is generated from a random variable. UAT modifies both the input and output structures of VAE so as to take the full advantage of DNN-based approach with auxiliary features, a structure similar to those introduced in [7]–[12]. UAT provides robust latent variables which intervene the mapping between the noisy observed features and the phonetic target by using the distributive information of the clean feature estimates.

UAT trains the latent variable parameters according to the maximum likelihood (ML) criterion, similar to those used in the conventional UD framework. Our method tackles the limitations posed by the traditional Gaussian-based approaches by in-

corporating the VIF-based latent variable to the stochastic noisy-to-clean mapping scheme, hence successfully modeling input uncertainty.

The proposed technique, along with the conventional DNN-based techniques, is evaluated on Aurora-4 [16] and CHiME-4 databases [15]. Experimental results show that the proposed technique outperforms the conventional DNN-based techniques. Moreover, we confirm that the latent variables obtained from the proposed technique can be utilized as an effective measure of uncertainty.

5.2 Uncertainty Decoding for Noise Robustness

Under the general framework of HMM-based recognition, the likelihood $p^{(LH)}(\mathbf{y}_t|\mathbf{q}_t)$ with respect to a HMM state $\mathbf{q}_t$ given a noisy feature vector $\mathbf{y}_t$ can be written as follows:

$$p^{(LH)}(\mathbf{y}_t|\mathbf{q}_t) = \frac{p(\mathbf{q}_t|\mathbf{y}_t)p(\mathbf{y}_t)}{p(\mathbf{q}_t)}. \tag{5.1}$$

In (5.1), $p(\mathbf{q}_t|\mathbf{y}_t)$ and $p(\mathbf{q}_t)$ respectively represent the posterior and prior probabilities of $\mathbf{q}_t$ and $p(\mathbf{y}_t)$ is the prior probability density of $\mathbf{y}_t$ which does not influence the recognition process. In DNN-based acoustic model the posterior probability is usually given by

$$p(\mathbf{q}_t|\mathbf{y}_t) \cong f_{\mathbf{q}_t}(\mathbf{y}_{t-\tau:t+\tau}) \tag{5.2}$$

where $f_{\mathbf{q}_t}(\cdot)$ represents a mapping from the noisy features to the corresponding HMM state identity $\mathbf{q}_t$ implemented by a DNN. The subscript τ indicates the temporal coverage considered as the contextual information of the speech signal. The function $f_{\mathbf{q}_t}(\cdot)$ is directly learned based on a collection of noisy data in the multi-condition training scenario [10].

Moreover, if an auxiliary feature $\mathbf{a}_t$ is provided as an augmented input, (5.2) can be modified into

$$p(\mathbf{q}_t|\mathbf{y}_t) \cong f^{aux.}_{\mathbf{q}_t}(\mathbf{y}_{t-\tau:t+\tau}, \mathbf{a}_t) \tag{5.3}$$

where $f^{aux.}_{\mathbf{q}_t}(\cdot)$ represents a function predicting the corresponding phonetic target based on both the noisy input and auxiliary features. By applying this simple process to both training and decoding, the DNN can automatically learn the complicated mapping from the noisy speech possibly with the auxiliary features to the HMM state labels [10]–[12].

The feature-based techniques, on the other hand, map the noisy features into the corresponding clean features via a DNN and the obtained clean feature estimates are fed to the acoustic model. This can be described as

$$p(\mathbf{q}_t|\mathbf{y}_t) \cong p(\mathbf{q}_t|f_{\mathbf{x}}(\mathbf{y}_{t-\tau:t+\tau})) \tag{5.4}$$

where the output of $f_{\mathbf{x}}(\cdot)$ is a stream of clean feature estimates,

$$\hat{\mathbf{x}}_{t-\tau:t+\tau} = f_{\mathbf{x}}(\mathbf{y}_{t-\tau:t+\tau}). \tag{5.5}$$

In (5.4) and (5.5), $f_{\mathbf{x}}(\cdot)$ represents a function dealing with the mapping from the noisy to the clean speech features. As in the model-based techniques, the performance of a feature mapping technique can be improved with the incorporation of the auxiliary features [7].

Most of the DNN-based techniques [3]–[12], [60] aim to optimize the point estimates of the targets (e.g., clean features, and acoustic model parameters). Despite their success in robust ASR, the performance of these approaches usually degrades when there exist some mismatches between the training and test data. While the

training data set is limited to rather narrow environments, the test data may undergo distortions not observed in the training data. UD attempts to compensate the imperfection of the estimators in the decoding process.

In order to take account of the estimation errors, the UD provides a somewhat different view for the observation likelihood formulation. UD models begin by assuming that there does exist some training-test mismatch. In other words, UD assumes that the acoustic model is trained on clean speech data while the observed inputs are distorted versions of the underlying clean feature vectors [40]. And the mapping from the clean to the distorted feature is assumed to follow a stochastic process. Given the underlying assumptions, the observation likelihood can now be formulated via two steps: estimating the posterior densities of the clean features and marginalizing over the clean features. The likelihood is given by

$$p^{(LH)}(\mathbf{y}_t|\mathbf{q}_t) = p(\mathbf{y}_t)\int \frac{p(\mathbf{x}_t|\mathbf{q}_t)p(\mathbf{x}_t|\mathbf{y}_t)}{p(\mathbf{x}_t)}d\mathbf{x}_t \tag{5.6}$$

where we assume that $p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{q}_t) \approx p(\mathbf{y}_t|\mathbf{x}_t)$. Then, by formulation, the influence of unreliable estimates can be de-emphasized in the decoding process.

The toughest challenge in implementation of the UD techniques resides in marginal integration as found on the RHS of (5.6), since it is often computationally intractable. Conventionally, such issues were tackled by approximating the likelihood using Gaussian or Gaussian mixture densities. In a GMM-HMM system, all three density terms, i.e., $p(\mathbf{x}_t|\mathbf{q}_t), p(\mathbf{x}_t|\mathbf{y}_t)$ and $p(\mathbf{x}_t)$ on the RHS of Equation (5.6), are assumed to be a Gaussian or Gaussian mixtures.

In contrast, DNN-HMM based UD techniques allows numerical evaluation of the integral [54]–[57]. The posterior probability $p(\mathbf{q}_t|\mathbf{y}_t)$ is obtained by computing the expectation of clean posterior $p(\mathbf{q}_t|\mathbf{x}_t)$ over the estimated distribution of clean

feature, i.e.,

$$p(\mathbf{q}_t|\mathbf{y}_t) \cong \mathbb{E}[p(\mathbf{q}_t|\mathbf{x}_t)|\hat{\mathbf{x}}_t, b^2_{\mathbf{x}_t}] \tag{5.7}$$

where $b^2_{\mathbf{x}_t}$ represents the variance-related term of $\mathbf{x}_t$. The process estimating $\hat{\mathbf{x}}_t$ and $b^2_{\mathbf{x}_t}$ is called uncertainty estimation. In general, additional speech enhancement techniques, such as nonnegative matrix factorization or Wiener filter, are employed to compute $\hat{\mathbf{x}}_t$ during the uncertainty estimation process [57]. Similarly, it is common to obtain $b_{\mathbf{x}_t}$ using some heuristic approaches or approximation methods, yet this sometimes causes performance degradation of the overall UD process [42], [43], [45], [48]. Once the information of the clean feature distribution is given, the expectation of the clean posterior is approximately computed by the numerical sampling. This process is referred to as uncertainty propagation.

Although uncertainty estimation and propagation may serve as options for resolving the uncertainty issues, there is a tradeoff with increased computational burdens. When the tradeoff between performance and computation burden is too costly, then the current UD framework may not be the most optimal choice for the DNN-HMM structure when utilizing uncertainty in the DNN acoustic model. Such issues call for a new method in lieu of the existing ones when dealing with uncertainty.

5.3 Variational Autoencoder

VAE is a generative model which combines the idea from an autoencoder with statistical inference. One of the most important characteristics of the VAE is that it can perform efficient approximate inference in the presence of continuous latent variables with intractable posterior distributions [58]. Through the stochastic gradient variational Bayes (SGVB) algorithm, the VAE parameters are optimized to

carry out an efficient posterior inference without the need for expensive iterative inference schemes (e.g., Markov Chain Monte Carlo).

Analogous to the model structure of the standard autoencoder, the VAE is composed of two directed networks: encoder and decoder networks. However, unlike the standard autoencoder, the VAE assumes that the observed data $\mathbf{o}$ is generated by a Gaussian random variable $\mathbf{z}$ which has normal prior distribution. The encoder network of the VAE outputs the mean and covariance of the posterior Gaussian distribution given the input and the decoder network tries to reconstruct the input pattern from that information.

In the mathematical perspective of the VAE framework, the network parameters are trained to maximize the likelihood given an observation $\mathbf{o}$, which is given by

$$p_\theta(\mathbf{o}) = \int p_\theta(\mathbf{z}) p_\theta(\mathbf{o}|\mathbf{z}) d\mathbf{z} \tag{5.8}$$

where θ represents the generative parameters of the VAE. The integral in the RHS of (5.8) usually becomes intractable when the generative model $p_\theta(\mathbf{o}|\mathbf{z})$ is implemented by a deep structured neural network. This also makes the true posterior density $p_\theta(\mathbf{z}|\mathbf{o}) = p_\theta(\mathbf{o}|\mathbf{z})p_\theta(\mathbf{z})/p_\theta(\mathbf{o})$ intractable. Therefore, it is hard to evaluate or differentiate the marginal likelihood using the general neural network training framework.

In order to alleviate this problem, the VAE framework introduces $q_\phi(\mathbf{z}|\mathbf{o})$, a probabilistic function which provides a variational approximation of the intractable true posterior $p_\theta(\mathbf{z}|\mathbf{o})$ with ϕ denoting the variational parameters. Here, the encoder specifying $q_\phi(\mathbf{z}|\mathbf{o})$ approximates the posterior probability $p_\theta(\mathbf{z}|\mathbf{o})$ given an observation $\mathbf{o}$. The decoder implements $p_\theta(\mathbf{o}|\mathbf{z})$ by reconstructing $\mathbf{o}$ from the latent variable generated by the encoder network.

VAE parameters are trained to maximize the log-likelihood given a training sample $\mathbf{o}$ which can be written as follows [58]:

$$\log p_\theta(\mathbf{o}) = D_{KL}(q_\phi(\mathbf{z}|\mathbf{o})||p_\theta(\mathbf{z}|\mathbf{o})) + \mathscr{L}(\theta, \phi; \mathbf{o}). \tag{5.9}$$

The first term in the RHS of (5.9) represents the Kullback-Leibler divergence (KL divergence) between the approximated posterior $q_\phi(\mathbf{z}|\mathbf{o})$ and the true posterior $p_\theta(\mathbf{z}|\mathbf{o})$. Since the KL divergence is non-negative, the second term in the RHS of (9) becomes the variational lower bound on the log-likelihood and it is given by

$$\begin{aligned} \log p_\theta(\mathbf{o}) \geq & \mathscr{L}(\theta, \phi; \mathbf{o}) \\ = & - D_{KL}(q_\phi(\mathbf{z}|\mathbf{o})||p_\theta(\mathbf{z})) \\ & + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{o})}[\log p_\theta(\mathbf{o}|\mathbf{z})]. \end{aligned} \tag{5.10}$$

The encoder and the decoder networks of the VAE can be trained jointly by maximizing the variational lowerbound $\mathscr{L}(\theta, \phi; \mathbf{o})$ with respect to ϕ and θ. The first term in the RHS of (5.10) represents the KL divergence between the prior and the posterior distributions of the latent variable $\mathbf{z}$, which acts as a regularization penalty. Since both distributions are described as Gaussians the KL divergence term has a simple closed form determined in terms of their parameters.

The second term in the RHS of (5.10), i.e., the expectation of conditional log-likelihood $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{o})}[\log p_\theta(\mathbf{o}|\mathbf{z})]$ means the reconstruction error between the input and output of the VAE. This can be approximated by the reparameterization trick which computes the Monte Carlo (MC) estimate of the variational lower bound. Based on the assumption that the prior and posterior densities are Gaussian with diagonal

covariance matrices, $\mathscr{L}(\theta, \phi; \mathbf{o})$ is now given by

$$\begin{aligned}\mathscr{L}(\theta, \phi; \mathbf{o}) = \frac{1}{2}\sum_{d=1}^{D}(1 + \log(\sigma_{\mathbf{z}_d})^2 - (\mu_{\mathbf{z}_d})^2 + (\sigma_{\mathbf{z}_d})^2) \\ + \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(\mathbf{o}|\mathbf{z}^{(l)})\end{aligned} \tag{5.11}$$

where $\mathbf{z}^{(l)} \sim q_\phi(\mathbf{z}|\mathbf{o})$ and D denote the dimensionality of $\mathbf{z}$, and $\mu_{\mathbf{z}_d}$ and $\sigma_{\mathbf{z}_d}$ are respectively the d-th element of the posterior mean and standard deviation of $\mathbf{z}$, i.e., $\mu_{\mathbf{z}}$ and $\sigma_{\mathbf{z}}$. Also, L indicates the number of samples used for estimation and the l-th sample $\mathbf{z}^{(l)}$ can be reparameterized as

$$\mathbf{z}^{(l)} = \mu_{\mathbf{z}} + \sigma_{\mathbf{z}}\epsilon \tag{5.12}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is an auxiliary noise variable. Summarizing, the VAE training attempts to minimize the reconstruction error while maximizing the similarity between the prior and posterior distributions of the latent variable. Interested readers are referred to [58] for more detail on VAE.

5.4 VIF-based uncertainty-aware Training

In this section, we propose a new VIF-based uncertainty-aware training technique, UAT, namely, which systematically measures and takes account of the uncertainty inherent in the input features by using a single deep neural network. UAT provides a novel network design substituting the existing uncertainty modules in the current UD approaches for DNN-HMM. More specifically, UAT uses individual DNNs for uncertainty estimation and propagation, employing various DNN-based techniques and VIF. For implementation, we begin by introducing the DNN-based

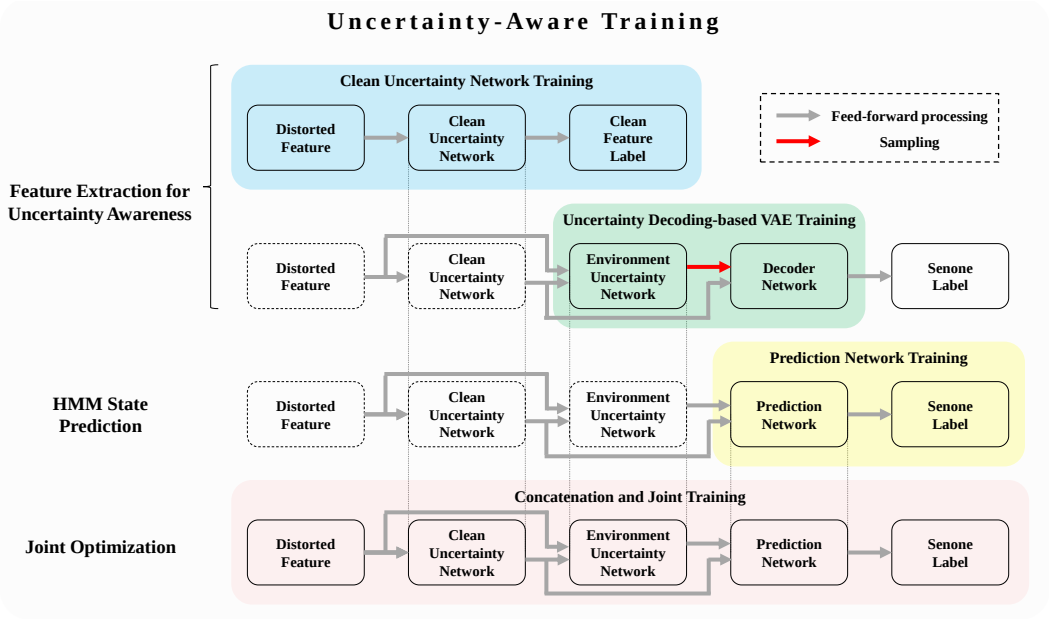


Figure 5.1: The training procedure of uncertainty-aware training.

acoustic models to the conventional UD scheme. In this case, (5.6) is modified as follows:

$$p^{(LH)}(\mathbf{y}_t|\mathbf{q}_t) = p(\mathbf{y}_t)\frac{\int p(\mathbf{q}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_t)d\mathbf{x}_t}{p(\mathbf{q}_t)} \tag{5.13}$$

where $p(\mathbf{x}_t|\mathbf{q}_t)$ and $p(\mathbf{x}_t)$ are respectively substituted by $p(\mathbf{q}_t|\mathbf{x}_t)$ and $p(\mathbf{q}_t)$ since the DNN-based acoustic model provides the posterior probability of the states. In (5.13), $p(\mathbf{y}_t)$ can be treated as a constant and we remove it from the likelihood formulation for simplicity.

As pointed out in the previous section, marginal integration over the clean feature is almost intractable when using DNN. In this study, we propose a DNN struc-

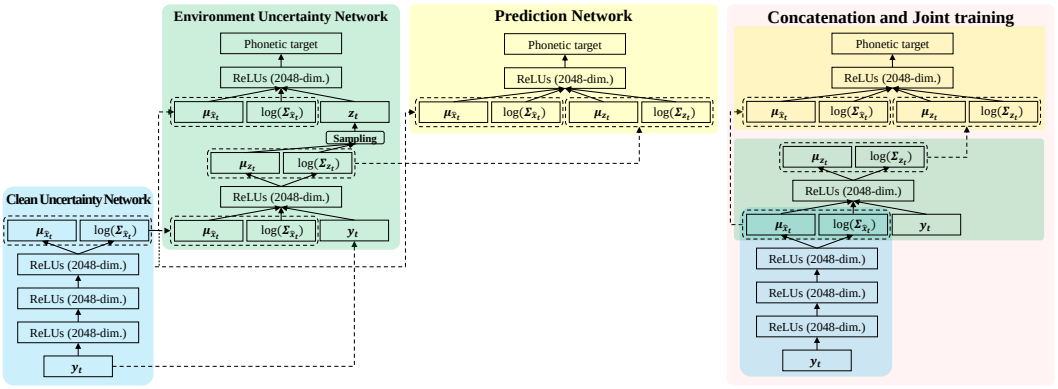


Figure 5.2: The network structure of uncertainty-aware training.

ture designed specifically to tackle this intractability issue. When we analyze (5.13), we can see that the likelihood with a corrupted feature $\mathbf{y}_t$ is determined by the conditional distribution $p(\mathbf{x}_t|\mathbf{y}_t)$. The conditional distribution $p(\mathbf{x}_t|\mathbf{y}_t)$ enables the likelihood to account for the uncertainty originating from the distorted input feature. In the conventional feature-based techniques, a point estimate $\widehat{\mathbf{x}}_t$ is derived from $\mathbf{y}_t$, i.e., $p(\mathbf{x}_t|\mathbf{y}_t) = \delta(\mathbf{x}_t - \widehat{\mathbf{x}}_t)$, and it can be clearly seen that it does not consider any distributional characteristics of the enhanced feature. Meanwhile, if $p(\mathbf{x}_t|\mathbf{y}_t)$ is specified as a parametric distribution, the likelihood should depend on its parameters, too. Especially, among the parameters, the variance-related terms of the distribution which represent the reliability of the estimated mean is directly related to the input uncertainty.

Therefore, under the assumption that the clean feature $\mathbf{x}_t$ given the noisy feature $\mathbf{y}_t$ follows a certain parametric distribution, the RHS of (5.13) can be defined as a

function of the parameters of $p(\mathbf{x}_t|\mathbf{y}_t)$

$$p^{(LH)}(\mathbf{y}_t|\mathbf{q}_t) \cong g_{\mathbf{q}_t}(\xi_{\mathbf{x}_t}) \tag{5.14}$$

where, $g_{\mathbf{q}_t}(\cdot)$ represents a mapping from the parameters of $p(\mathbf{x}_t|\mathbf{y}_t)$ to the corresponding HMM state $\mathbf{q}_t$ and $\xi_{\mathbf{x}_t}$ denotes the parameters. By utilizing the capability of DNN to implement $g_{\mathbf{q}_t}$, we can implicitly build the complicated relationship between $\mathbf{y}_t$ and $\mathbf{q}_t$ including the marginalization with respect to $\mathbf{x}_t$. Analogous to this, if $\xi_{\mathbf{x}_t}$ is expressed as a function of $\mathbf{y}_t$, i.e., $\xi_{\mathbf{x}_t}(\mathbf{y}_t)$, then (5.14) can be described as a sequential two-stage DNN mapping, where the first stage carries out the parameter estimation of clean feature distribution, and the second stage, the subsequent HMM state prediction using the parameters resulting from the first stage.

However, there still exists some difficulty in this approach. Although the relationship between the distorted feature $\mathbf{y}_t$ and the corresponding HMM state $\mathbf{q}_t$ is described in terms of the intermediate distributional information of the clean feature $\mathbf{x}_t$ as defined in (5.13), its performance in adverse environment is still far worse when in adverse environment as compared to the clean conditions.

Since the conditional distribution $p(\mathbf{x}_t|\mathbf{y}_t)$ is estimated by a DNN from the training data, the aforementioned design still does not resolve the training-test data mismatch issues perfectly. Therefore, still required is the inclusion if additional latent variables which effectively accounts for the uncertainty coming from the unknown environment factors in the mapping between $\mathbf{y}_t$, $\mathbf{x}_t$ and $\mathbf{q}_t$.

We address above issue and supplement our UAT technique by introducing a latent variable $\mathbf{z}_t$ to Equation (5.13) as an additional environment feature that intervenes the mapping from the corrupted feature $\mathbf{y}_t$ to the corresponding state $\mathbf{q}_t$, which the clean feature $\mathbf{x}_t$ cannot explain. Now (5.13) is modified to incorporate the

environment latent variable $\mathbf{z}_t$ as follows:

$$p^{(LH)}(\mathbf{y}_t|\mathbf{q}_t) \propto \frac{\int\int p(\mathbf{q}_t|\mathbf{x}_t,\mathbf{z}_t)p(\mathbf{x}_t,\mathbf{z}_t|\mathbf{y}_t)d\mathbf{x}_t d\mathbf{z}_t}{p(\mathbf{q}_t)}$$
$$= \frac{\int\int p(\mathbf{q}_t|\mathbf{x}_t,\mathbf{z}_t)p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{y}_t)p(\mathbf{x}_t|\mathbf{y}_t)d\mathbf{x}_t d\mathbf{z}_t}{p(\mathbf{q}_t)}. \tag{5.15}$$

It can be seen from (5.15) that $p^{(LH)}(\mathbf{y}_t|\mathbf{q}_t)$ is determined not only by the clean feature but also by the latent variable. If $p(\mathbf{x}_t|\mathbf{y}_t)$ and $p(\mathbf{z}_t|\mathbf{x}_t,\mathbf{y}_t)$ are parameterized with parameters $\xi_{\mathbf{x}_t}$ and $\xi_{\mathbf{z}_t}$, respectively, then we have

$$p^{(LH)}(\mathbf{y}_t|\mathbf{q}_t) \cong g^{sup.}_{\mathbf{q}_t}(\xi_{\mathbf{x}_t},\xi_{\mathbf{z}_t}) \tag{5.16}$$

where $g^{sup.}_{\mathbf{q}_t}(\cdot)$ is the function predicting the target $\mathbf{q}_t$ based on the parameters $\xi_{\mathbf{x}_t}$ and $\xi_{\mathbf{z}_t}$. The relationship between (5.14) and (5.16) is similar with that between (5.2) and (5.3). As the auxiliary feature $\mathbf{a}_t$ in (5.3) supplements the mapping from the noisy input feature $\mathbf{y}_t$ to the corresponding HMM state $\mathbf{q}_t$ by providing an information which $\mathbf{y}_t$ cannot consider, the parameters of the latent variable $\xi_{\mathbf{z}_t}$ in (5.16) assist the clean feature parameters $\xi_{\mathbf{x}_t}$ with the prediction of $\mathbf{q}_t$, especially well under the training-test mismatch condition.

In order to implement (5.16), the UAT technique carries out two tasks: feature extraction for uncertainty awareness and HMM state prediction. The former task involves derivation of $\xi_{\mathbf{x}_t}$ and $\xi_{\mathbf{z}_t}$ given an input feature $\mathbf{y}_t$. Then, the during the HMM state prediction phase, the posterior probability estimation over the HMM states given the input containing not only the point estimates of $\mathbf{x}_t$ and $\mathbf{z}_t$ but also the parameters of their posterior distributions is performed based on $g^{sup.}_{\mathbf{q}_t}(\xi_{\mathbf{x}_t},\xi_{\mathbf{z}_t})$ shown in (5.16). This allows the likelihood $p^{(LH)}(\mathbf{y}_t|\mathbf{q}_t)$ to be characterized by the distribution of both $\mathbf{x}_t$ and $\mathbf{z}_t$ conditioned on $\mathbf{y}_t$ so as to take advantage of the uncertainty of the estimation.

In this work, we assume that both the conditional distributions, $p(\mathbf{x}_t|\mathbf{y}_t)$ and $p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{y}_t)$ are given by Gaussian pdfs where each component of $\mathbf{x}_t$ and $\mathbf{z}_t$ is uncorrelated.

$$p(\mathbf{x}_t|\mathbf{y}_t) = \mathcal{N}(\mathbf{x}_t; \mu_{\mathbf{x}_t}(\mathbf{y}_t), \Sigma_{\mathbf{x}_t}(\mathbf{y}_t)) \tag{5.17}$$

$$p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{y}_t) = \mathcal{N}(\mathbf{z}_t; \mu_{\mathbf{z}_t}(\mathbf{x}_t, \mathbf{y}_t), \Sigma_{\mathbf{z}_t}(\mathbf{x}_t, \mathbf{y}_t)). \tag{5.18}$$

We should note that $\mu_{\mathbf{x}_t}$ and $\Sigma_{\mathbf{x}_t}$ depend on $\mathbf{y}_t$ while $\mu_{\mathbf{z}_t}$ and $\Sigma_{\mathbf{z}_t}$ depend on both $\mathbf{x}_t$ and $\mathbf{y}_t$. In this parametric formulation $\xi_{\mathbf{x}_t} = \{\mu_{\mathbf{x}_t}, \Sigma_{\mathbf{x}_t}\}$ and $\xi_{\mathbf{z}_t} = \{\mu_{\mathbf{z}_t}, \Sigma_{\mathbf{z}_t}\}$, respectively. In the proposed technique, $p(\mathbf{x}_t|\mathbf{y}_t)$ is computed by a neural network which we call the clean uncertainty network (CUN), and $p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{y}_t)$ is derived from the VAE of modified structure which we call the environment uncertainty network (EUN). In this paper, the clean uncertainty represents the uncertainty appearing in the process of clean feature estimation which has the same meaning of the conventional meaning of the uncertainty. On the other hand, the environment uncertainty means the uncertainty that cannot be fully resolved by the CUN due to the some unseen factors. Hence, by definition, the two types of the uncertainties are complementary of each other and fully encompasses the sources of uncertainties especially in the training-test mismatch condition. The input of CUN is the noisy feature $\mathbf{y}_t$ and the output is $\xi_{\mathbf{x}_t} = \{\mu_{\mathbf{x}_t}, \Sigma_{\mathbf{x}_t}\}$. The parameters of CUN are trained to maximize the log-likelihood $\log p(\mathbf{x}_t|\mathbf{y}_t)$.

Now we apply the modified VAE framework to model $p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{y}_t)$ in (5.18). In order to model $p(\mathbf{x}_t|\mathbf{y}_t)$, the practical implementation of the VAE calls for an approximation of (5.15) into a simplified form. Meanwhile, the parametric information of $p(\mathbf{x}_t|\mathbf{y}_t)$ has already been given by CUN. Therefore, we take an advantage of

the given parameters of $\mathbf{x}_t$, i.e., $\xi_{\mathbf{x}_t}$, for modeling $p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{y}_t)$. EUN replaces the marginalization over $\mathbf{x}_t$ in (5.15) by a parametric dependency on $\xi_{\mathbf{x}_t}$ instead. Based on this, (5.15) may be approximated as following:

$$\begin{aligned} p^{(LH)}(\mathbf{y}_t|\mathbf{q}_t) &\propto \frac{\int p_\theta(\mathbf{q}_t|\xi_{\mathbf{x}_t}, \mathbf{z}_t) q_\phi(\mathbf{z}_t|\xi_{\mathbf{x}_t}, \mathbf{y}_t) d\mathbf{z}_t}{p(\mathbf{q}_t)} \\ &= \frac{\int p_\theta(\mathbf{q}_t|\mu_{\mathbf{x}_t}, \Sigma_{\mathbf{x}_t}, \mathbf{z}_t) q_\phi(\mathbf{z}_t|\mu_{\mathbf{x}_t}, \Sigma_{\mathbf{x}_t}, \mathbf{y}_t) d\mathbf{z}_t}{p(\mathbf{q}_t)} \end{aligned} \tag{5.19}$$

where θ and ϕ represent the parameters of the decoder and encoder, respectively. Here, $q_\phi(\mathbf{z}_t|\mu_{\mathbf{x}_t}, \Sigma_{\mathbf{x}_t}, \mathbf{y}_t)$ takes over the role played by $p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{y}_t)$.

In order to combine the above processes in a single network, the training procedure and network structure of the UAT are designed as in Figs. 5.4 and 5.4. The UAT network is composed of three individually trained DNNs and ultimately fine-tuned via joint training. The first network, CUN derives $\mu_{\mathbf{x}_t}(\mathbf{y}_t)$ and $\Sigma_{\mathbf{x}_t}(\mathbf{y}_t)$ from the given noisy input feature $\mathbf{y}_t$ as in (5.17). The second network, EUN corresponds to the encoder network of the UD-based VAE which computes $\mu_{\mathbf{z}_t}(\mathbf{x}_t, \mathbf{y}_t)$ and $\Sigma_{\mathbf{z}_t}(\mathbf{x}_t, \mathbf{y}_t)$ in (5.18) with an approximation shown in (5.19). The last DNN predicts the posterior probability corresponding to each HMM state, i.e., $g^{sup.}_{\mathbf{q}_t^*}(\xi_{\mathbf{x}_t}, \xi_{\mathbf{z}_t})$ in (5.16), which we call the prediction network (PN).

5.4.1 Clean Uncertainty Network

The main goal of CUN is to derive the parametric information of the clean feature distribution given the noisy input feature $p(\mathbf{x}_t|\mathbf{y}_t)$. The objective function of the network for training CUN can be formulated as follows:

$$J_{CUN} = \frac{1}{T}\sum_{t=1}^{T} \log p(\mathbf{x}_t|\mathbf{y}_t), \tag{5.20}$$

with

$$\log p(\mathbf{x}_t|\mathbf{y}_t) = \sum_{d=1}^{D_\mathbf{x}} -\log\left(\sqrt{2\Sigma_{\widehat{\mathbf{x}}_{t,d}}\pi}\right) - \frac{(\mathbf{x}_{t,d} - \mu_{\widehat{\mathbf{x}}_{t,d}})^2}{2\Sigma_{\widehat{\mathbf{x}}_{t,d}}} \tag{5.21}$$

where $\mu_{\widehat{\mathbf{x}}_{t,d}}$ and $\Sigma_{\widehat{\mathbf{x}}_{t,d}}$ are the d-th elements of $\mu_{\widehat{\mathbf{x}}_t}$ and $\Sigma_{\widehat{\mathbf{x}}_t}$, respectively. Also, $D_\mathbf{x}$ denote the dimensionality of $\mathbf{x}_t$. The output vector $\mathbf{o}_t^{CUN}$ of CUN is given by:

$$\mathbf{o}_t^{CUN} = [\mu_{\widehat{\mathbf{x}}_t}', \log(\Sigma_{\widehat{\mathbf{x}}_t})']' \tag{5.22}$$

where

$$\Sigma_{\widehat{\mathbf{x}}_t} = [\Sigma_{\widehat{\mathbf{x}}_{t,1}}', \Sigma_{\widehat{\mathbf{x}}_{t,2}}', ..., \Sigma_{\widehat{\mathbf{x}}_{t,D_\mathbf{x}}}']'. \tag{5.23}$$

While the mean term, i.e., $\mu_{\widehat{\mathbf{x}}_t}$, sets the clean feature as its target, the variance term, i.e., $\Sigma_{\widehat{\mathbf{x}}_t}$, does not require a specific target. The variance term is computed autonomously as CUN minimizes the objective function given $\mathbf{x}_t$, the ultimate target of the mean term. That is, in other words, CUN is structured so as to be able to generate the variance terms without needing any target or heuristics.

5.4.2 Environment Uncertainty Network

In the training state of EUN, we slightly modify the conventional VAE in order to successfully model the latent variable $\mathbf{z}_t$. The encoder of the modified VAE computes $q_\phi(\mathbf{z}_t|\mu_{\mathbf{x}_t}, \Sigma_{\mathbf{x}_t}, \mathbf{y}_t)$ in (5.19) and the decoder tries to estimate the posterior probability of the corresponding HMM state $\mathbf{q}_t$ given the reparameterized latent variable $\mathbf{z}_t$ generated by the encoder network, i.e., EUN, and the clean feature parameters. The reparameterization of $\mathbf{z}_t$ is performed by sampling from the posterior distribution $\mathcal{N}(\mathbf{z}_t; \mu_{\mathbf{z}_t}(\mathbf{x}_t, \mathbf{y}_t), \Sigma_{\mathbf{z}_t}(\mathbf{x}_t, \mathbf{y}_t))$ as follows [58]:

$$\mathbf{z}_t = \mu_{\mathbf{z}_t} + \sigma_{\mathbf{z}_t}\epsilon \tag{5.24}$$

where $\sigma_{\mathbf{z}_t}$ is the standard deviation of the latent variable.

The input vector $\mathbf{v}_t^{EUN}$ of the encoder of the EUN and that of the decoder $\mathbf{v}_t^{DEC.}$ are

$$\mathbf{v}_t^{EUN} = [\mathbf{y}_t{}', \mu_{\widehat{\mathbf{x}}_t}{}', \log(\Sigma_{\widehat{\mathbf{x}}_t})']' \tag{5.25}$$

$$\mathbf{v}_t^{DEC.} = [\mathbf{z}_t{}', \mu_{\widehat{\mathbf{x}}_t}{}', \log(\Sigma_{\widehat{\mathbf{x}}_t})']'. \tag{5.26}$$

As shown in (5.25) and (5.26), conditioning the input of both the encoder and decoder networks on the parametric information of the clean speech features, the VAE can successfully model the latent variables which assist the clean features with mapping the noisy features to the corresponding HMM states [61], [62].

The objective function of the modified VAE, i.e., EUN is given by

$$\begin{aligned} J_{EUN} =& D_{KL}(q_\phi(\mathbf{z}_t|\mu_{\widehat{\mathbf{x}}_t}, \Sigma_{\widehat{\mathbf{x}}_t}, \mathbf{y}_t)||p(\mathbf{z}_t|\mu_{\widehat{\mathbf{x}}_t}, \Sigma_{\widehat{\mathbf{x}}_t})) \\ & - \frac{1}{L}\sum_{l=1}^{L} \log(p_\theta(\mathbf{q}_t|\mu_{\widehat{\mathbf{x}}_t}, \Sigma_{\widehat{\mathbf{x}}_t}, \mathbf{z}_t^{(l)})). \end{aligned} \tag{5.27}$$

The first RHS terms in (5.10) and (5.27) are almost identical except that $\mathbf{o}$ from (5.10) is substituted with $\mathbf{v}_t^{EUN}$ in (5.27). The second term on the RHS of (5.27) represents the cross-entropy between the output vector of the decoder and the corresponding HMM state. Therefore, the modified VAE is trained to maximize not only the estimated posterior probability but also the similarity between the prior and the posterior distributions of the latent variable. Optimizing this objective function leads EUN to extract the environmental uncertainty of the input in the process of mapping to the phonetic target based on the UD. The conditional prior distribution $p(\mathbf{z}_t|\mu_{\widehat{\mathbf{x}}_t}, \Sigma_{\widehat{\mathbf{x}}_t})$ is assumed to follow $\mathcal{N}(0, \mathbf{I})$ as in the standard VAE [58], [61], [63], [64].

5.4.3 Prediction Network and Joint Training

Once the training of CUN and EUN is completed, PN computes $g_{\mathbf{q}_t}^{sup.}(\xi_{\mathbf{x}_t}, \xi_{\mathbf{z}_t})$ in (5.16). The input vector of PN $\mathbf{v}_t^{PN}$ is given by

$$\mathbf{v}_t^{PN} = [\mu_{\widehat{\mathbf{x}}_t}', \log(\Sigma_{\widehat{\mathbf{x}}_t})', \mu_{\mathbf{z}_t}', \log(\Sigma_{\mathbf{z}_t})']'. \tag{5.28}$$

By using the variance terms of both $\mathbf{x}_t$ and $\mathbf{z}_t$ as the input without any additional process (e.g., MC sampling), we can let the acoustic modeling to consider the uncertainty of clean speech and environment information simultaneously.

After PN is optimized, CUN, EUN, and PN are concatenated together to form a unified single DNN. Then, the unified network is trained jointly according to the cross-entropy criterion. Specifically, the error signal between the output of the unified DNN and the corresponding phonetic target flows back to PN, EUN and CUN, and consequently trains all the parameters. With this series of processes, learning the relationship between the noisy features and the corresponding HMM state can be enhanced by guiding the DNN through the intermediate level features, i.e., the parametric information of the clean estimates and the latent variables.

5.5 Experiments

In this section, we present a series of experiments on two different tasks: Aurora-4 and CHiME-4 databases. The detailed information for Aurora-4 and CHiME-4 DBs is given in 2.2.1 and 2.2.4, respectively.

In order to verify the performance of the proposed technique, several DNN-based acoustic modeling techniques were implemented and their performances were compared with that of the proposed UAT. In addition to the ASR performance eval-

uations of all the DNN-based techniques, we analyze whether the proposed system can represent the uncertainty inherent in the observed inputs effectively.

5.5.1 Experimental Setup: Feature Extraction and ASR System

We used the Kaldi speech recognition toolkit [31] for feature extraction, GMM-HMM training, alignment, and ASR decoding. Meanwhile, all the DNN-based techniques were implemented by Keras [65] and trained using the ADADELTA optimization technique [66]. Also, dropout [20] with a fraction of 0.2 and L2 regularization with a weight of 0.00002 were applied for training all the networks.

The two GMM-HMM systems were trained uniformly by the standard recipe of Kaldi. 13-dimensional MFCCs (including C_0), as well as their first- and second-order derivatives, were processed using linear discriminant analysis (LDA) with a context size of 7 frames (i.e., ± 3) and maximum likelihood linear transform (MLLT) sequentially. The numbers of senones in the Aurora-4 and CHiME-4 were 2006 and 1987, respectively. As for the language model, we applied the standard 5k open tri-gram. The tri-gram in CHiME-4 DB, meanwhile, was rescored by a RNN language model which is provided as one of the baseline language models in CHiME.

Feature extraction for the DNN techniques was performed by the default configuration of Kaldi. For Aurora-4 DB, mean-normalized 24-dimensional LMFB features including their first- and second-order derivatives were used as input of all the networks. In the case of the CHiME-4 DB, we used the same configuration while variance normalization was performed additionally.

5.5.2 Network Structures

The performance of the proposed technique was compared with five different methods of acoustic modeling incorporating the various DNN-based techniques for robust ASR. The compared techniques are

- *DNN-Baseline*: Multi-condition DNN-HMM,
- *DNN-Conventional*: Conventional DNN-based acoustic modeling using the clean feature estimates as intermediate features [60],
- *DNN-ID*: Standard DNN-based acoustic modeling with the same structure to UAT,
- *VAE-Conventional*: Conventional VAE-based acoustic modeling [67],
- *DNN-UD-MC*: Conventional uncertainty decoding technique for DNN-based acoustic modeling based on MC sampling [54], [55],
- *UAT*: Proposed UAT.

Details on the architecture of the techniques, except for *DNN-Baseline* are provided in Fig. . The input layer of all the models had a total of 792 visible units obtained by windowing 11 consecutive LMFB features, i.e., τ was set to be 5. Also, all the models had 7 hidden layers and a softmax output layer where each unit corresponds to a senone, and each hidden layer of *DNN-Baseline* consisted of 2048 rectified linear units (ReLUs).

All the techniques except for *DNN-Baseline* attempts to guide the mapping from the observed input to the corresponding HMM state via each of the intermediate feature layers. As shown in Fig. 5.4 and described in Section 5.4, *UAT* exploits the

mean and variance terms of $\mathbf{x}_t$ and $\mathbf{z}_t$ as the intermediate features at the sixth hidden layer. From the perspective of practical implementation of the parametric representation of $\mathbf{x}_t$, it is more beneficial for the CUN to consider the contextual coverage of the observed input $\mathbf{y}_{t-\tau:t+\tau}$. In the experiments, we modify $\mathbf{o}_t^{CUN}$ and $\mathbf{v}_t^{PN}$ in (5.22) and (5.28) as follows:

$$\mathbf{o}_t^{CUN} = [\mu_{\widehat{\mathbf{x}}_{t-\tau:t+\tau}}', \log(\Sigma_{\widehat{\mathbf{x}}_{t-\tau:t+\tau}})']' \tag{5.29}$$

$$\mathbf{v}_t^{PN} = [\mu_{\widehat{\mathbf{x}}_{t-\tau:t+\tau}}', \log(\Sigma_{\widehat{\mathbf{x}}_{t-\tau:t+\tau}})', \mu_{\mathbf{z}_t}', \log(\Sigma_{\mathbf{z}_t})']' \tag{5.30}$$

where

$$\begin{aligned} \mu_{\widehat{\mathbf{x}}_{t-\tau:t+\tau}} &= [\mu_{\mathbf{x}_{t-\tau}}', ..., \mu_{\mathbf{x}_{t+\tau}}']' \\ \Sigma_{\widehat{\mathbf{x}}_{t-\tau:t+\tau}} &= [\Sigma_{\mathbf{x}_{t-\tau}}', ..., \Sigma_{\mathbf{x}_{t+\tau}}']' \end{aligned} \tag{5.31}$$

to take a longer contextual window into consideration. CUN applied in *UAT* and *DNN-ID* was composed of 3 hidden layers with 2048 ReLU nodes and an output layer with a total of 1584 linear units including $\mu_{\widehat{\mathbf{x}}_{t-\tau:t+\tau}}$ and $\log(\Sigma_{\widehat{\mathbf{x}}_{t-\tau:t+\tau}})$ of 792 dimensions, respectively. For the VAE-based techniques, we ran the experiment with latent variables whose dimensions were 128 and 256.

Among the various techniques compared in this experiment, *DNN-Conventional* is the representative conventional DNN-based technique employing the deterministic estimates of the clean features without any source of uncertainty. *DNN-Conventional* exploits the clean feature estimates, i.e, the output of the conventional feature enhancement network $\widehat{\mathbf{x}}_{t-\tau:t+\tau}$, as the intermediate features at the fourth hidden layer. The enhancement network performs the role of $f_{\mathbf{x}}(\cdot)$ in (5.5). In this paper, we refer the enhancement network as clean deterministic network (CDN). CDN was trained according to the minimum mean squared error (MMSE) criterion and outputs the corresponding 792-dimensional clean feature estimates.

DNN-ID was included in the experiment in order to check and compare the effect of EUN in *UAT*. *DNN-ID* had an identical structure with that of *UAT* but used a standard DNN instead of a VAE, i.e., $\mu_{\mathbf{z}_t}$ and $\log(\Sigma_{\mathbf{z}_t})$ of *UAT* at the sixth layer were substituted with ReLUs of the same dimensionality, i.e., 256 and 512. By comparing the results of *UAT* to that of *DNN-ID*, we assess how the latent variable parameters of *UAT* supplement the environment information.

At the same time, we compare the results of *UAT* to that of the *VAE-Conventional* in order to assess whether the effectiveness of the latent variable representation varies dependent on different modeling approaches. *VAE-Conventional* trains parameters by maximizing the marginal log-likelihood of the observed features. In this process, *VAE-Conventional* lacks any other resources such as the clean features and phonetic targets in the training. Such an approach may result in different representations for the latent variables in *VAE-Conventional* as compared to those in *UAT*.

Finally, we include *DNN-UD-MC* in our experiment in order to test competitiveness of *UAT* against the existing UD approach of DNN-HMM structures employing uncertainty propagation based on the MC sampling [54]. The implementation of *DNN-UD-MC* originates from (5.7). In order to assure fair comparison, CDN was employed for estimating $\hat{\mathbf{x}}_t$ instead of speech enhancement model (e.g., Wiener filter). In the estimation of $b^2_{\mathbf{x}_t}$, Delcroix's uncertainty (DU) estimator which is one of most widely known techniques for uncertainty estimation was used [55]. The DU estimator is obtained by assuming the uncertainty to be proportional to the squared difference between the enhanced features $\hat{\mathbf{x}}_t$ and the noisy features $\mathbf{y}_t$:

$$b^2_{\mathbf{x}_t} = \alpha(\mathbf{y}_t - \hat{\mathbf{x}}_t)^2 \tag{5.32}$$

where α was set 0.8 by the a series of experiments checking the best ASR perfor-

Table 5.1: Comparison of averaged Euclidean distance between the clean feature targets and the unprocessed inputs, Gaussian means of CUN and outputs of CDN over the test set.

Database	Unprocessed	CUN	CDN
Aurora-4	0.290	0.233	0.231
CHiME-4	0.204	0.150	0.147

mance. Once the uncertainty estimation is given, the uncertainty propagation is carried out by utilizing the MC sampling as following:

$$\mathbb{E}[p(\mathbf{q}_t|\mathbf{x}_t)|\hat{\mathbf{x}}_t, b^2_{\mathbf{x}_t}] = \frac{1}{L}\sum_{l=1}^{L} \log p(\mathbf{q}_t|\mathbf{x}_t{}^{(l)}) \tag{5.33}$$

where l-th sample of $\mathbf{x}_t$, $\mathbf{x}_t{}^{(l)}$ can be reparameterized as

$$\mathbf{x}_t{}^{(l)} = \hat{\mathbf{x}}_t + b_{\mathbf{x}_t}\epsilon. \tag{5.34}$$

L was set 20.

Mini-batch size for the ADADELTA algorithm was set to 512 for all the DNNs. The learning rate was set to be 1 for training all the networks, except for the cases of joint training where the learning rate was set to be 0.1. Training of each network was stopped after 20 epochs.

5.5.3 Effects of CUN on the Noise Robustness

We first examined the noise robustness of CUN by plotting the trajectories of its output over an utterance in the test sets of Aurora-4 and CHiME-4 DBs. Figs. 5.6-(a) and (c) display the trajectories of the clean, noisy features, clean feature estimates

obtained from CDN, and Gaussian means of the clean feature estimates obtained from CUN on Aurora-4 and CHiME-4 DBs, respectively. Both the deterministic estimates and the Gaussian means follow the clean feature trajectory except for the cases when the Euclidean distance between the noisy features and the clean features is large. In order to compare the accuracy of the deterministic estimates and Gaussian means, we calculated the average Euclidean distances between Gaussian means of CUN and the corresponding clean feature targets, as well as the clean targets and outputs of CDN, over the test sets of the two databases. The result is reported in Table 5.1. Results show that the difference between the Gaussian mean and deterministic estimate is almost negligible regardless of the databases.

Figs. 5.6-(b) and (d) plot the noise features and the log-variances of the clean estimates obtained from CUN trained on Aurora-4 and CHiME-4 databases, respectively. Since the variance term is representative of the reliability of the estimated value, we claim it to increase as the distortion level of the input feature grows higher. The results indeed agree with our claim, from which we may conclude that the output of CUN contains additional uncertainty-related information which CDN cannot provide.

5.5.4 Uncertainty Representation in Different SNR Condition

As mentioned in the previous section, *UAT* utilizes the variance of the latent variable as an index of environment uncertainty. Now we test whether the latent variable parameters of *UAT* are in fact effective representation of the environment uncertainty. More specifically, we compute the frame-wise differential entropies of the latent variables.

The differential entropy is one of the most classical measures of the uncertainty

of a continuous random variable. Since the latent variable $\mathbf{z}_t$ from the VAEs follows a Gaussian distribution, the differential entropy may be computed in the following:

$$H(\mathbf{z}_t) = \frac{1}{2}\log(2\pi e)^{D_{\mathbf{z}}} + \frac{1}{2}\log\prod_{d=1}^{D_{\mathbf{z}}}\Sigma_{\mathbf{z}_{t,d}} \tag{5.35}$$

where $D_{\mathbf{z}}$ denotes the dimensionality of $\mathbf{z}_t$. Also, $\Sigma_{\mathbf{z}_{t,d}}$ is the d-th element of $\Sigma_{\mathbf{z}_t}$.

For each test set of Aurora-4 DB and SIMU in CHiME-4 DB, we calculated the differential entropy of the latent variable with the dimension of 128 inferred from the two VAE-based techniques, *UAT* and *VAE-Conventional*. We also computed the differential entropy of CUN outputs, i.e., $\mathbf{o}_t^{CUN}$ as it can serve as a measure of the input uncertainty. *DNN-UD-MC* was excluded in the differential entropy comparison, since its clean distribution cannot be described as Gaussian. Because the dimensionality of $\mathbf{o}_t^{CUN}$ is different from that of the latent variable, the entropy is scaled by the dimensionality of each variable.

We averaged the differential entropies obtained from six different segmental SNR (SSNR) groups. SSNR is one of commonly-used speech quality measures, which computes the average of the SNR values of short segments instead of the entire signal. By calculating the differential entropy separately by each SSNR group, we can now analyze the effect of speech deterioration on uncertainty in the feature estimation process. Mathematically speaking, SSNR is computed as follows:

$$SSNR = \frac{10}{M}\sum_{m=0}^{M-1}\log_{10}\frac{\sum_{i=Nm}^{Nm+N-1}y^2(i)}{\sum_{i=Nm}^{Nm+N-1}(y(i)-x(i))^2}, \tag{5.36}$$

where $y(i)$ and $x(i)$ are the noisy and clean speech samples indexed by i, and N and M are the segment length and the number of segments respectively. In this work, we set $N = 128$ and $M = 11$.

The differential entropies on Aurora-4 DB and CHiME-4 DB were averaged into 6 different SSNR groups: higher than 12 dB, 12∼-9 dB, 9∼-6 dB, 6∼-3 dB, 3∼-0

dB and lower than 0 dB groups and higher than -1 dB, -1∼-2 dB, -2∼-3 dB, -3∼-4 dB, -4∼-5 dB and lower than 5 dB groups. Averaged entropies were scaled to fit the dynamic range between 0 and 1 according to each technique. Figs. 5.6-(a) and 5-(b) present the histograms of the differential entropies on two databases, computed using the outputs of CUNs. The plots show that the scaled average entropies were somewhat irregular across different SSNR groups. Especially in the case of CHiME-4 DB, it can be seen that the differential entropy decreases slightly as SSNR value increases. Graphical observations lead to the conclusion that SSNR is not the dominant feature in determining the clean uncertainty.

The differential entropies obtained from the latent variables of the two VAE-based techniques, *UAT* and *VAE-Conventional*, display tendencies different from the differential entropies from CUN. While the entropy of *VAE-Conventional* shows little variation, *UAT* clearly tends to increase as the SSNR condition becomes poorer. The relative increment in the differential entropy of the VAE-based techniques between the first group and the sixth group obtained from *VAE-Conventional* and *UAT* were 2.33% and 36.12% in Aurora-4 DB and -0.69% and 14.36% in CHiME-4 DB, respectively. The results lead to a conclusion that the methods adopted by the UAT framework to model the latent variable show superior performance in terms of capturing uncertainty as compared to those used in *VAE-Conventional*. Moreover, we observe encouraging results from which we may assume that CUN and EUN complement each other in a sense that each network shows distinctively different responses under certain environmental conditions.

On the other hand, in order to test EUN's effectiveness in providing the unseen environment condition information as a form of supplementary feature which CUN cannot directly capture, we check whether the results of EUN effectively represent

different environment conditions frame-by-frame. However, since the environment uncertainty is a latent feature that cannot be observed directly, finding an appropriate measure for it still remains a challenge. This issue can be addressed by observing the variations of EUN outputs with respect to the test set samples which contain different environmental distortions. More specifically, we focus on the variations of the degrees of training-test mismatch given the entire test set samples with various environment distortions rather than examining the output value of a specific sample. The degree of training-test mismatch is measure frame-by-frame, and the variation of the EUN output is computed frame-wise. Here, we define the training-test mismatch as the disparity between the training and test data sets invoked by environmental factors of unseen patterns. It is nevertheless a difficult task to correctly quantify the degree of mismatch since it is not observed directly, so as environment factors are not. Hence, we approximate the degree of mismatch by measuring the intensity of some phenomenon in the aftermath of training-test mismatch.

The degradation of clean estimate accuracy is one of the foremost phenomenon arising due to the training-test mismatch. Meanwhile, Fig. 5.6 shows that the distance between the noisy and the clean estimate increases more as the gap between the clean and clean estimates grows larger. Such a result provides grounds for using the Euclidean distance between the noisy and the clean estimate feature as the measure of the degree of mismatch.

In order to check whether the EUN outputs effectively represent environment uncertainty with respect to the chosen mismatch measure, we applied PCA to the supervectors including the mean and the log-variance of the latent variables, i.e., $[\mu_{\mathbf{z}_t}{}', \log(\Sigma_{\mathbf{z}_t})']'$, and visualized the results. For comparison with other techniques, PCA was applied to the latent variable parameters of *VAE-Conventional* and CUN

outputs. Fig. 5.6 visualizes the two resulting dominant components, mean- and variance-normalized, which are plotted separately for 4 different Euclidean distance groups of CHiME-4 SIMU and REAL. For the sake of visual interpretability, we randomly selected 2000 samples for each database. In both CHiME-4 SIMU and REAL, it can be easily seen that the distribution of the EUN outputs is shifted slightly according to the Euclidean distance, of which the result is clearly in contrast to those of the latent variable parameters of *VAE-Conventional* and CUN outputs. Particularly, when the distances between the red and cyan dots, i.e., representing the poorest and the best Euclidean condition respectively, are compared, the distribution of EUN outputs is evident from those of the others clearly. It may be interpreted that the latent variable parameters of EUN provide distinctive information. From this, we conclude that EUN of the proposed technique may potentially provide supplementary information about the environment uncertainty which CUN and conventional VAE technique fail to provide, especially under the training-test mismatch condition.

5.5.5 Result of Speech Recognition

Table 5.2 and 5.4 list performance on the ASR tasks tested on Aurora-4 and CHiME-4 DBs using all of the acoustic modeling techniques in the comparison group. The results show that *UAT* reports the best performance for every test condition. It is especially encouraging that *UAT* improves performance in regards to the artificial noise as well as the real noise originally unobserved in the training set (i.e., REAL). In contrast, *DNN-UD-MC* performs poorly as compared to other techniques. Although it shows a light improvement from the *DNN-Baseline*, it may not be sufficient enough of an improvement so as to compensate the increase in the computational cost. With the consideration of the tradeoff between computational

cost and model performance, then, UAT framework proves to be quite competitive against the existing DNN-based UD framework in the scope of uncertainty control. On the other hand, *VAE-Conventional* shows the lowest performance. This helps explaining why UAT showed persistently better results as compared to those of the VAE-based techniques in terms of the latent variable performance in various aspects within the scope of uncertainty capturing as reported in Figs. 5.6 and 5.6.

Additionally, we computed relative error rate reductions (RERRs) for all of the techniques in the comparison group. Compared with the conventional DNN-based techniques including *DNN-Conventional* and *DNN-ID*, the relative error rate reductions (RERRs) of *UAT (128)* are 10.66% and 9.74% in Aurora-4 DB. Also, in SIMU and REAL the RERRs of *UAT (256)* over *DNN-Conventional* and *DNN-ID-512* are 10.61% and 9.36%, and 13.36% and 8.99%, respectively. This result shows that, in overall, the proposed UAT technique outperforms the conventional DNN-based techniques.

5.5.6 Result of Speech Recognition with LSTM-HMM

From the previous ASR experiments, the proposed UAT technique has shown better performance in various environment conditions. This is due to the fact that the outputs of CUN and EUN play a useful role in the DNN-based acoustic model. As an extension, we have conducted experiments where UAT is applied to RNN-based acoustic models. The underlying assumption here is that, if the uncertainties modeled by CUN and EUN are fed to the RNN-based acoustic model covering sufficient size of context information in a sequential manner, the advantages of identifying and including the two different uncertainties in the training may be amplified.

More specifically, we applied the UAT framework to long-short term memory

Table 5.2: WERs (%) on the compared acoustic modeling techniques on Aurora-4 testset.

Method(L.V. Dim.)	A	B	C	D	Avrg.
DNN-Baseline	3.12	7.43	7.33	17.84	11.58
DNN-Conventional	2.97	6.60	6.13	16.81	10.69
DNN-ID-256	2.91	6.62	6.14	16.63	10.61
DNN-ID-512	2.87	6.56	6.46	16.57	10.58
VAE-Conventional (128)	2.96	7.41	7.33	18.66	11.91
VAE-Conventional (256)	3.00	7.52	7.44	18.94	12.09
DNN-UD-MC	3.01	7.07	6.98	17.06	11.05
UAT (128)	2.62	**5.86**	5.72	**15.03**	**9.55**
UAT (256)	**2.59**	5.92	**5.70**	15.11	9.61

(LSTM)-HMM [68], one of the state-of-the-art acoustic model on CHiME-4 DB. We call this technique *LSTM-UAT*. In order to compare the performance of *LSTM-UAT*, we also implemented *LSTM-Baseline* and *LSTM-ID* which are the basic multi-condition LSTM-HMM and the LSTM-HMM version of *DNN-ID* technique introduced in the previous subsections, respectively. *LSTM-Baseline* had 3 layers of 1024 memory cells. We specifically chose *LSTM-ID* as one of the comparison models, since its DNN-analogous, *DNN-ID*, shows the most competitive results against the rest of the comparison techniques in the previous performance tests. In order to ensure fair comparison in terms of the number of parameters, *LSTM-UAT* and *LSTM-ID* were derived from *UAT (128)* and *DNN-ID (256)*, respectively. *LSTM-UAT* and

Table 5.3: WERs (%) on the compared acoustic modeling techniques on CHiME-4 testset.

Method(L.V. Dim.)	SIMU	REAL
DNN-Baseline	12.84	20.66
DNN-Conventional	12.35	19.99
DNN-ID-256	12.27	19.25
DNN-ID-512	12.18	19.03
VAE-Conventional (128)	13.71	20.75
VAE-Conventional (256)	13.79	20.91
DNN-UD-MC	12.54	20.43
UAT (128)	11.19	17.43
UAT (256)	**11.04**	**17.32**

LSTM-ID replace individual PNs of *UAT (128)* and *DNN-ID (256)* with 3 layers of 1024 memory cells. The output state label was delayed by 5 frames. The network structures of the two LSTM-based techniques except *LSTM-Baseline* are shown in Fig. 5.6. One distinctive property of the two LSTM-based techniques is that they only take in the t-th element of $\mathbf{o}_t^{CUN}$, while maintaining the 256-dimensional EUN output and the ReLUs. This assures that the input vector structure of LSTM-based PN is identical to $\mathbf{v}_t^{PN}$ in (5.27).

Meanwhile, LSTM requires data of greater volume than DNN for training. In this context, the size of CHiME-4 DB's training set is moderately small for optimization via LSTM-HMM system. We addressed this issue by employing the training data

Table 5.4: Computation complexity measurement of the compared acoustic modeling techniques.

Method(L.V. Dim.)	No. of param.	xRT
DNN-Baseline	30.9 M	0.021
DNN-Conventional	24.8 M	0.013
DNN-ID-256	24.6 M	0.013
VAE-Conventional (128)	23.6 M	0.011
DNN-UD-MC	24.8 M	0.263
UAT (128)	24.6 M	0.013

Table 5.5: WERs (%) on the compared LSTM-based acoustic modeling techniques on CHiME-4 testset.

Method(L.V. Dim.)	SIMU	REAL
LSTM-Baseline	10.76	16.68
LSTM-ID-256	10.13	15.63
LSTM-UAT (128)	**9.25**	**14.17**

from all channels available for training LSTM-part of our proposed model. Such an approach is referred to as multi-microphone training [69]. The multi-microphone training is known to enhance the robustness of LSTM-based acoustic models in response environment variability by feeding more abundant data for training.

In order to train these LSTMs, truncated backpropagation through time (BPTT) was used. As in the DNN-based techniques, ADADELTA was again used for the

Table 5.6: Computation complexity measurement of the compared LSTM-based acoustic modeling techniques.

Method(L.V. Dim.)	No. of param.	xRT
LSTM-Baseline	23.3 M	0.10
LSTM-ID-256	40.0 M	0.14
LSTM-UAT (128)	40.0 M	0.12

LSTM optimization and the other settings for training the LSTMs such as mini-batch size and learning rate remained identical as set for the DNN training. Note that regularization settings such as dropout [20] and L2 regularization were not retained when training the LSTMs. Also, the training of all the LSTMs including joint training of the unified networks was stopped after 10 epochs.

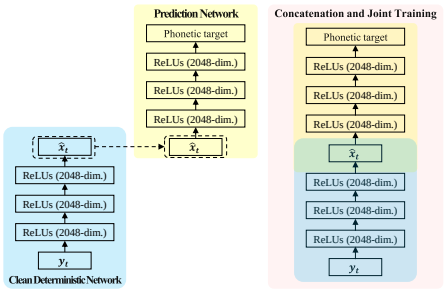
As shown in Table 5.6, *LSTM-UAT (128)* was better than *LSTM-Baseline* and *LSTM-ID-256* in ASR performance in both tested conditions. In SIMU and REAL, the RERRs of *LSTM-UAT (128)* over *LSTM-Baseline* and *LSTM-ID-256* were 14.03% and 8.69%, and 15.05% and 9.34%, respectively. The reported results suggest that the parametric information of the clean and environment estimates of the UAT framework performs well with a network whose structure involves sequential training.

5.6 Summary

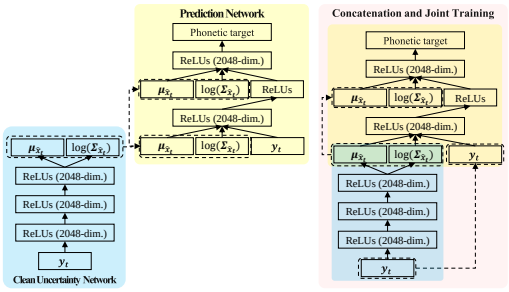
In this paper, a novel deep learning-based acoustic modeling technique for uncertainty awareness was proposed. In order to consider the input uncertainty in the

decoding process, the proposed technique employed a modified structure of VAE for modeling the robust latent variables which mediate the mapping between the noisy observed features and the phonetic target. The VAE of the proposed technique was trained according to the maximum likelihood (ML) criterion which was driven from the uncertainty framework optimized to be used with the deep learning-based acoustic models. The latent variable variances were employed as the uncertainty measure of the input along with the distributive information of the clean feature estimates.

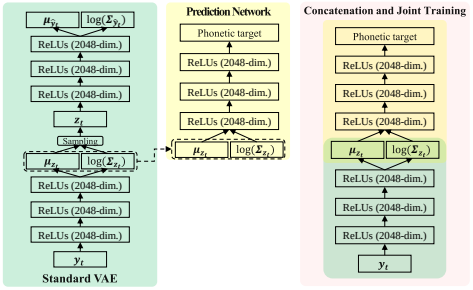
To evaluate the performance of our proposed technique, Aurora-4 and CHiME-4 databases were used. From the experimental results, we observed that the latent variables of the proposed technique effectively represent the level of uncertainty according to the SSNR and the clean uncertainty conditions. Moreover, we compared the performance of our proposed technique with the DNN-based technique of the identical network structure in two kinds of back-end model, i.e., DNN-HMM and LSTM-HMM.



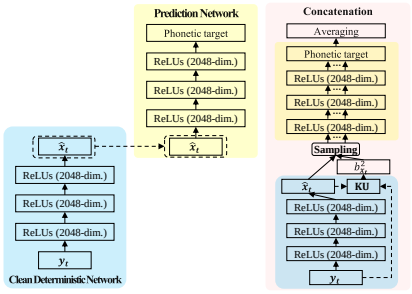
1) ***DNN-Conventional***



2) ***DNN-ID***



3) ***VAE-Conventional***



4) ***DNN-UD-MC***

Figure 5.3: The network structures and training procedures of compared techniques except for *DNN-Baseline*.

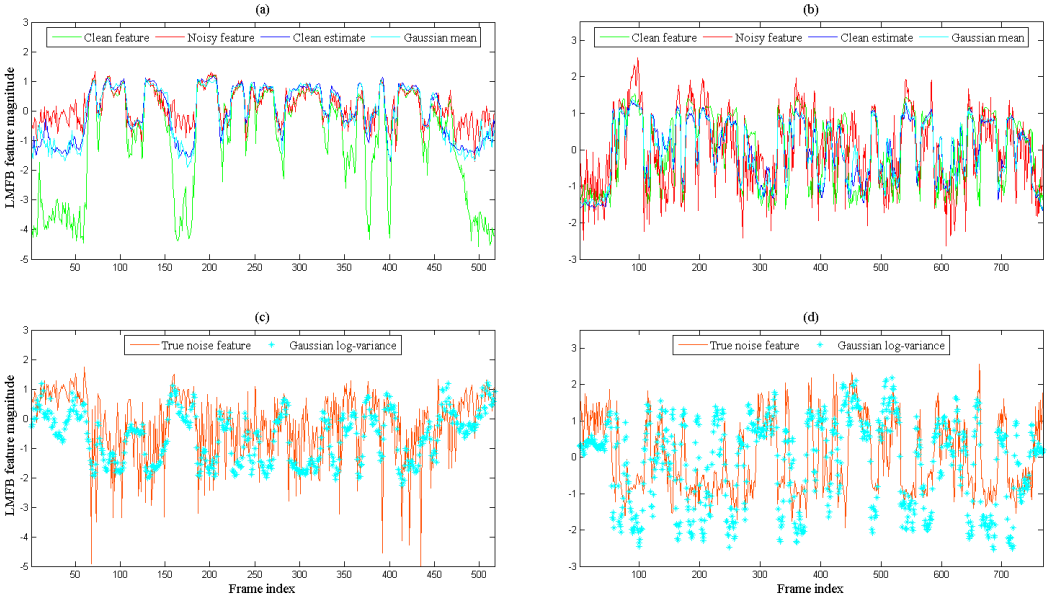


Figure 5.4: Effects of CUN. Trajectories of the 0-th LMFB features of clean, observed noisy speech, clean estimates, and Gaussian means of clean estimates on (a) Aurora-4 DB (b) CHiME-4 DB. Trajectories of the 0-th LMFB features of noise and log-variances of clean estimates on (c) Aurora-4 DB (d) CHiME-4 DB.

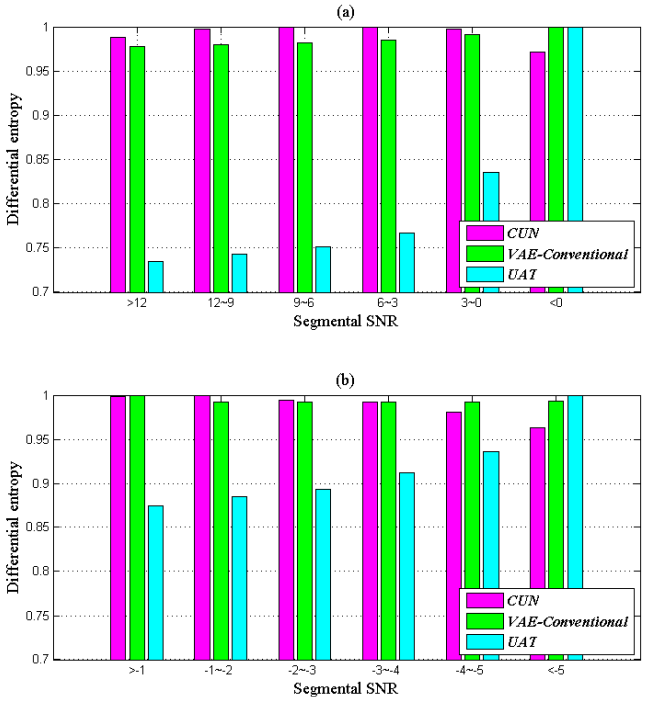


Figure 5.5: Average differential entropy computed using the variance of the latent variables and the clean estimates extracted from the various VAE-based acoustic modeling techniques and CUN on (a) Aurora-4 and (b) CHiME-4 databases, respectively.

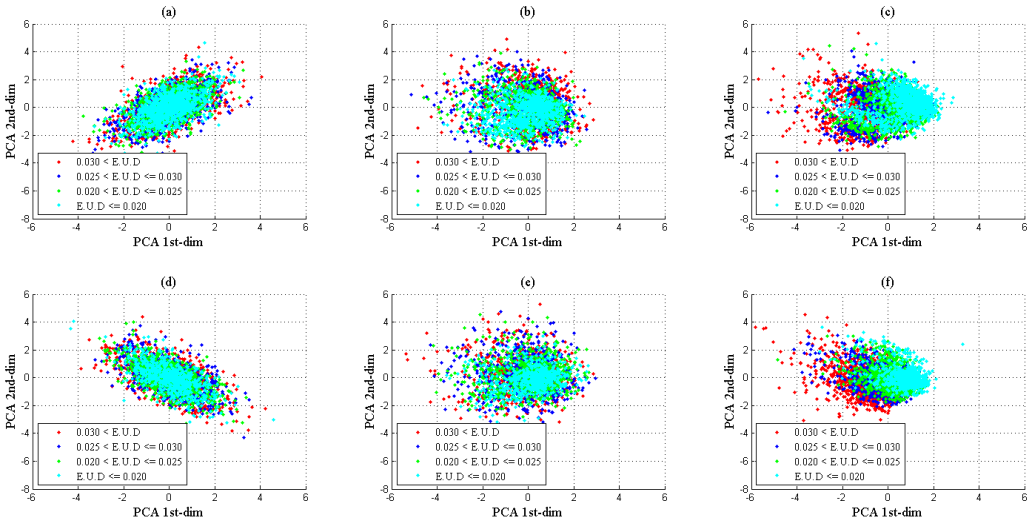
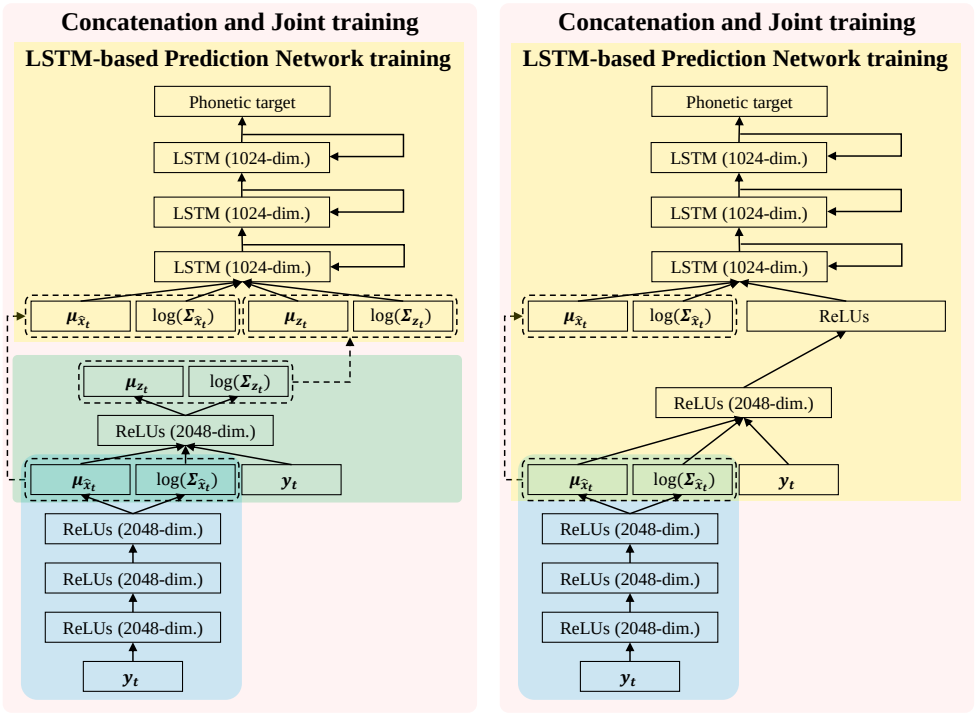


Figure 5.6: PCA projections of the latent variable supervectors of two VAE-based techniques on the Euclidean distance (E.U.D). The distributions of CUN output on SIMU (a) and REAL (d), *VAE-Conventional* on SIMU (b) and REAL (e), and those of *UAT* on SIMU (c) and REAL (f).



LSTM-UAT ***LSTM-ID***

Figure 5.7: The network structures of *LSTM-UAT* and *LSTM-ID*.

Chapter 6

Conclusions

In this thesis, the n model-based and data-driven approaches for the environment-robust speech recognition have been proposed. The sequential characteristic of the speech was modeled by HMM. According to the way to calculate the emission probabilities of the HMM, the type of the speech recognition decoder was divided into GMM-HMM and DNN-HMM systems. In the GMM-HMM system, the acoustic model was trained using the clean-condition training data and model-based technique was proposed in order to match the reverberant noisy input features with the characteristic of the trained acoustic model. In the DNN-HMM system, the DNN was trained using the multi-condition training data to obtain the relationship between the input and the target labels. In accordance with these concepts, we proposed four techniques for the environment-robust speech recognition.

In this thesis, four novel DNN-based acoustic modeling techniques for robust automatic speech recognition have been proposed. Firstly, we have proposed a DNN-based acoustic model designed for effective usage of multi-condition data and its noise estimate has been proposed. The proposed technique dealt with the mapping

from noisy speech and noise estimates to phonetic targets by concatenating two fine-tuned DNNs and training the unified network jointly. Through a series of experiments on Aurora-5 task and mismatched noise conditions, it has been shown that the proposed technique outperforms NAT in word accuracy on both matched and mismatched conditions.

Secondly, we have proposed a DNN-based feature enhancement approach for multichannel distant speech recognition. The proposed approach built a multichannel-based feature mapping DNN using conventional beamformer, DNN and its joint training technique with lapel microphone data. Through a series of experiments on MC-WSJ-AV corpus, we have found that the proposed technique clarifies the relationship between the features obtained from distant microphone array and clean speech.

Finally, a deep learning-based acoustic modeling technique for uncertainty awareness has been proposed. In order to consider the input uncertainty in the decoding process, the proposed technique employed a modified structure of VAE for modeling the robust latent variables which mediate the mapping between the noisy observed features and the phonetic target. The VAE of the proposed technique was trained according to the maximum likelihood (ML) criterion which was driven from the uncertainty framework optimized to be used with the deep learning-based acoustic models. The latent variable variances were employed as the uncertainty measure of the input along with the distributive information of the clean feature estimates. To evaluate the performance of our proposed technique, Aurora-4 and CHiME-4 databases were used. From the experimental results, we observed that the latent variables of the proposed technique effectively represent the level of uncertainty according to the SSNR and the clean uncertainty conditions. Moreover, we compared

the performance of our proposed technique with the DNN-based technique of the identical network structure in two kinds of back-end model, i.e., DNN-HMM and LSTM-HMM.

Bibliography

[1] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[2] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[3] A. Narayanan and D. Wang, "Investigation of speech separation as a front-end for noise robust speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 4, pp. 826–835, Apr. 2014.

[4] W. Li, Y. Z. L. Wang, J. Dines, M. Magimai.-Doss, H. Bourlard, and Q. Liao, "Feature mapping of multiple beamformed sources for robust overlapping speech recognition using a microphone array," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 12, pp. 2244–2255, Dec. 2014.

[5] X. Feng, Y. Zhang, and J. Glass, "Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition," in *Proc. ICASSP*, 2014, pp. 1759–1763.

[6] M. Mimura, S. Sakai, and T. Kawahara, "Exploring deep neural networks and deep autoencoders in reverberant speech recognition," in *HSCMA*, 2014, pp. 197–201.

[7] K. H. Lee, W. H. Kang, T. G. Kang, and N. S. Kim, "Integrated DNN-based model adaptation technique for noise-robust speech recognition," in *Proc. ICASSP*, 2017, pp. 5245–5249.

[8] A. Narayanan and D. Wang, "Improving robustness of deep neural network acoustic models via speech separation and joint adaptive training," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 1, pp. 92–101, Jan. 2015.

[9] Z. Wang and D. Wang, "A joint training framework for robust autiomatic speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 4, pp. 796–806, Jan. 2016.

[10] M. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. ICASSP*, 2013, pp. 7398–7402.

[11] G. Saon, H. Nahamoo, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proc. ASRU*, 2013, pp. 55–59.

[12] M. Delcroix, K. Kinoshita, T. Hori, and T. Nakatami, "Context adaptive deep neural networks for fast acoustic model adaptation in noisy conditions," in *Proc. ICASSP*, 2016, pp. 5270–5274.

[13] G. Hirsch, "AURORA-5 experimental framework for the performance evaluation of speech recognition in case of a hands-free speech input in noisy environments," Tech. Rep., 2007.

[14] M. Lincoln, I. McCowan, J. Vepa, and H. K. Maganti, "The multi-channel wall street journal audio visual corpus (MC-WSJ-AV): specification and initial experiments," in *Proc. ASRU*, 2005, pp. 357–262.

[15] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "HMM adaptation using vector taylor series for noisy speech recognition," *Comput. Speech Lang.*, vol. 46, no. 11, pp. 537–557, Nov. 2000.

[16] G. Hirsch, "Experimental framework for the performance evaluation of speech recognition front-ends on a large vocabulary task, version 2.0," Tech. Rep., 2002.

[17] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum bayes risk training of deep neural network acoustic models using distributed hessian-free optimization," in *Proc. Interspeech*, 2012, pp. 10–13.

[18] D. Yu, M. L. Seltzer, J. Li, J. Huang, and F. Seide, "Feature learning in deep neural networks - A study on speech recognition tasks," *CoRR*, vol. abs/1301.3605, 2013.

[19] S. Young, "The HTK book," Tech. Rep., 2006.

[20] N. Srivastava, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, Jun. 2014.

[21] G. Hu. (2004) 100 nonspeech environmental sounds. [Online]. Available: http://web.cse.ohio-state.edu/pnl/corpus/HuNonspeech/HuCorpus.html

[22] J. Benesty, J. Chen, and Y. Huang, *Microphone array signal processing.* Springer, 2008.

[23] O. L. Frost, "An algorithm for linearly constrained adaptive array processing," in *Proc. IEEE*, vol. 60, no. 8, Aug. 1972, pp. 926–935.

[24] L. J. Griffiths and C. W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Trans. Antennas Propag.*, vol. AP-30, no. 1, pp. 27–34, Jan. 1982.

[25] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 32, no. 6, pp. 1109–1121, Dec. 1984.

[26] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, and A. Acero, "Robust speech recognition using a cepstral minimum-meansquare-error-motivated noise suppressor," *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, no. 5, pp. 1061–1070, Jul. 2008.

[27] N. S. Kim, "IMM-based estimation for slowly evolving environments," *IEEE Signal Process. Lett.*, vol. 5, no. 6, pp. 146–149, Jun. 1998.

[28] L. Deng, J. Droppo, and A. Acero, "Recursive estimation of nonstationary noise using iterative stochastic approximation for robust speech recognition," *IEEE Speech Audio Process.*, vol. 11, no. 6, pp. 568–580, Nov. 2003.

[29] T. Kowaliw, N. Bredeche, and R. Doursat, *Growing adaptive machines: combining development and learning in artificial neural networks.* Springer, 2014.

[30] T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals, "WSJCAM0: a british english speech corpus for large vocabulary continuous speech recognition," in *Proc. ICASSP*, 1995, pp. 81–84.

[31] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[32] A. Ghoshal and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. Interspeech*, 2013, pp. 2345–2349.

[33] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[34] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.

[35] J. A. Arrowood and M. A. Clements, "Using observation uncertainty in HMM decoding," in *Proc. Interspeech*, 2002, pp. 1561–1564.

[36] N. B. Yoma and M. Villar, "Speaker verification in noise using a stochastic version of the weighted viterbi algorithm," *IEEE Speech Audio Process.*, vol. 10, no. 3, pp. 158–166, May 2002.

[37] L. Deng, J. Droppo, and A. Acero, "Dynamic compensation of HMM variances using the feature enhancement uncertainty computed from a parametric model of speech distortion," *IEEE Speech Audio Process.*, vol. 13, May 2006.

[38] Q. Huo and C.-H. Lee, "A bayesian predictive classification approach to robust speech recognition," *IEEE Speech Audio Process.*, vol. 8, no. 2, pp. 200–204, Mar. 2003.

[39] V. Ion and R. Haeb-Umbach, "A novel uncertainty decoding rule with applications to transmission error robust speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, no. 5, pp. 1047–1060, Jul. 2008.

[40] D. Kolossa and R. Haeb-Umbach, in *Robust Speech Recognition of Uncertain or Missing Data: Theory and Applications.* Berlin: Springer-Verlag, 2011.

[41] H. Liao and M. Gales, "Joint uncertainty decoding for noise robust speech recognition," in *Proc. Interspeech*, 2005, pp. 3129–3132.

[42] M. Delcroix, T. Nakatani, and S. Watanabe, "Static and dynamic variance compsensation for recognition of reverberant speech with dereverberation preprocessing," *IEEE Trans. Audio, Speech, Language Process.*, vol. 17, no. 2, pp. 324–334, Jul. 2013.

[43] M. Delcroix, S. Watanabe, T. Nakatani, and A. Nakamura, "Cluster-based dynamic variance adaptation for interconnecting speech enhancement preprocessor and speech recognizer," *Comput. Speech Lang.*, vol. 27, no. 1, pp. 350–368, Jul. 2013.

[44] L. Lu, K. Chin, A. Ghoshal, and S. Renals, "Joint uncertainty decoding for noise robust subspace gaussian mixture models," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 9, pp. 1791–1804, Jul. 2013.

[45] D. Kolossa, R. F. Astudillo, E. Hoffmann, and R. Orglmeister, "Independent component analysis and time frequency masking for speech recognition in mul-

titalker condition," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2010, no. 1, pp. 1–13, Jul. 2010.

[46] R. F. Astudillo and R. Orglmeister, "Computing MMSE estimates and residual unertainty directly in the feature domain of ASR using STFT domain speech distortation models," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 5, pp. 1023–1034, May 2013.

[47] A. H. Abdelaziz, S. Zeiler, D. Kolossa, V. Leutnant, and R. Haeb-Umbach, "Gmm-based significance decoding," in *Proc. ICASSP*, 2013, pp. 6827–6831.

[48] F. Nesta, M. Matassoni, and R. F. Astudillo, "A flexible spatial blind source extraction framework for robust speech recognition in noisy environments," 2013, pp. 33–40.

[49] D. T. Tran, E. Vincent, and D. Jouvet, "Fusion of multiple uncertainty estimators and propagators for noise robust ASR," in *Proc. ICASSP*, 2014, pp. 5512–5516.

[50] ——, "Nonparametric uncertainty estimation and propagation for noise robust ASR," *IEEE Trans. Audio, Speech, Language Process.*, vol. 23, no. 11, pp. 1835–1846, Nov. 2015.

[51] A. Ozerov, M. Lagrange, and E. Vincent, "Uncertainty-based learning of acoustic models from noisy data," *Comput. Speech Lang.*, vol. 27, no. 3, pp. 874–894, Mar. 2013.

[52] R. F. Astudillo and J. P. da Silva Neto, "Propagation of uncertainty through multilayer perceptrons for robust automatic speech recognition," in *Proc. Interspeech*, 2011, pp. 461–464.

[53] R. F. Astudillo, A. Abad, and I. Trancoso, "Accounting for the residual uncertainty of multi-layer perceptron based features," in *Proc. ICASSP*, 2014, pp. 6859–6863.

[54] C. Huemmer, R. Mass, A. Schwarz, R. F. Astudillo, and W. Kellermann, "Uncertainty decoding for DNN-HMM hybrid systems based on numerical sampling," in *Proc. Interspeech*, 2015, pp. 3556–3560.

[55] A. H. Abdelaziz, S. Watanabe, J. R. Hershey, E. Vincent, and D. Kolossa, "Uncertainty propagation through deep neural networks," in *Proc. Interspeech*, 2015, pp. 3561–3565.

[56] C. Huemmer, A. Schwarz, R. Mass, H. Barfuss, R. F. Astudillo, and W. Kellermann, "A new uncertainty decoding scheme for DNN-HMM hybrid systems with multichannel speech enhancement," in *Proc. Interspeech*, 2016, pp. 5760–5764.

[57] K. Nathwani, E. Vincent, and I. Illina, "Consistent DNN uncertainty training and decoding for robust ASR," in *Proc. ASRU*, 2017, pp. 185–192.

[58] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. ICLR*, 2014.

[59] R. Salakhutdinov, "Learning deep generative models," *Annual Review of Statistics and Its Application*, vol. 2, no. 1, pp. 361–385, Jan. 2015.

[60] T. Gao, J. Du, L.-R. Dai, and C.-H. Lee, "Joint training of front-end and back-end deep neural networks for robust speech recognition," in *Proc. ICASSP*, 2015, pp. 4375–4379.

[61] C. Doersch, "Tutorial on variational autoencoders," in *arXiv:1606.05908*, 2016.

[62] H. L. K. Sohn and X. Yan, "Learning structured output representation using deep conditional generative models," in *Proc. NIPS*, 2015.

[63] H. Nishizaki, "Data augmentation and feature extraction using variational autoencoder for acoustic modeling," in *APSIPA*, 2017, pp. 1222–1227.

[64] S. Tan and K. C. Sim, "Learning utterance-level normalisation using variational autoencoders for robust automatic speech recognition," in *SLT*, 2016, pp. 3556–3560.

[65] F. Chollet. (2015) Keras. [Online]. Available: https://github.com/fchollet/keras

[66] M. D. Zeiler, "Adadelta: An adaptive learning rate method," in *arXiv preprint arXiv:1212.5701*, 2012.

[67] A. Tjandra, S. Sakti, S. Nakamura, and M. Adriani, "Stochastic gradient variational bayes for deep learning-based ASR," in *Proc. ASRU*, 2015, pp. 175–180.

[68] A. Graves, *Supervised Sequence Labeling with Recurrent Neural Networks.* ser. Studies in Computation Intelligence: Springer, 2012, vol. 385.

[69] T. Yoshioka, "The NTT CHiME-3 system: advances in speech enhancement and recognition for mobile multi-microphone devices," in *Proc. ASRU*, 2015, pp. 436–443.

요 약

본 논문에서는 강인한 음성인식을 위해서 DNN을 활용한 음향 모델링 기법들을 제안한다. 본 논문에서는 크게 세 가지의 DNN 기반 기법을 제안한다. 첫 번째는 DNN이 가지고 있는 잡음 환경에 대한 강인함을 보조 특징 벡터들을 통하여 최대로 활용하는 음향 모델링 기법이다. 이러한 기법을 통하여 DNN은 왜곡된 음성, 깨끗한 음성, 잡음 추정치, 그리고 음소 타겟과의 복잡한 관계를 보다 원활하게 학습하게 된다. 본 기법은 Aurora-5 DB 에서 기존의 보조 잡음 특징 벡터를 활용한 모델 적응 기법인 잡음 인지 학습 (noise-aware training, NAT) 기법을 크게 뛰어넘는 성능을 보였다.

두 번째는 DNN을 활용한 다 채널 특징 향상 기법이다. 기존의 다 채널 시나리오에서는 전통적인 신호 처리 기법인 빔포밍 기법을 통하여 향상된 단일 소스 음성 신호를 추출하고 그를 통하여 음성인식을 수행한다. 우리는 기존의 빔포밍 중에서 가장 기본적 기법 중 하나인 delay-and-sum (DS) 빔포밍 기법과 DNN을 결합한 다 채널 특징 향상 기법을 제안한다. 제안하는 DNN은 중간 단계 특징 벡터를 활용한 공동 학습 기법을 통하여 왜곡된 다 채널 입력 음성 신호들과 깨끗한 음성 신호와의 관계를 효과적으로 표현한다. 제안된 기법은 multichannel wall street journal audio visual (MC-WSJAV) corpus에서의 실험을 통하여, 기존의 다채널 향상 기법들보다 뛰어난 성능을 보임을 확인하였다.

마지막으로, 불확정성 인지 학습 (Uncertainty-aware training, UAT) 기법이다. 위에서 소개된 기법들을 포함하여 강인한 음성인식을 위한 기존의 DNN 기반 기법들은

각각의 네트워크의 타겟을 추정하는데 있어서 결정론적인 추정 방식을 사용한다. 이는 추정치의 불확정성 문제 혹은 신뢰도 문제를 야기한다. 이러한 문제점을 극복하기 위하여 제안하는 UAT 기법은 확률론적인 변화 추정을 학습하고 수행할 수 있는 뉴럴 네트워크 모델인 변화 오토인코더 (variational autoencoder, VAE) 모델을 사용한다. UAT는 왜곡된 음성 특징 벡터와 음소 타겟과의 관계를 매개하는 강인한 은닉 변수를 깨끗한 음성 특징 벡터 추정치의 분포 정보를 이용하여 모델링한다. UAT의 은닉 변수들은 딥 러닝 기반 음향 모델에 최적화된 uncertainty decoding (UD) 프레임워크로부터 유도된 최대 우도 기준에 따라서 학습된다. 제안된 기법은 Aurora-4 DB와 CHiME-4 DB에서 기존의 DNN 기반 기법들을 크게 뛰어넘는 성능을 보였다.

주요어: 강인한 음성인식, 특징 향상, 특징 보상, feature compensation,음향 모델링, acoustic modeling, 음향 모델 적응, acoustic model adaptation, deep neural network (DNN), variational autoencoder (VAE), variational inference, uncertainty decoding (UD)

학 번: 2012-20822